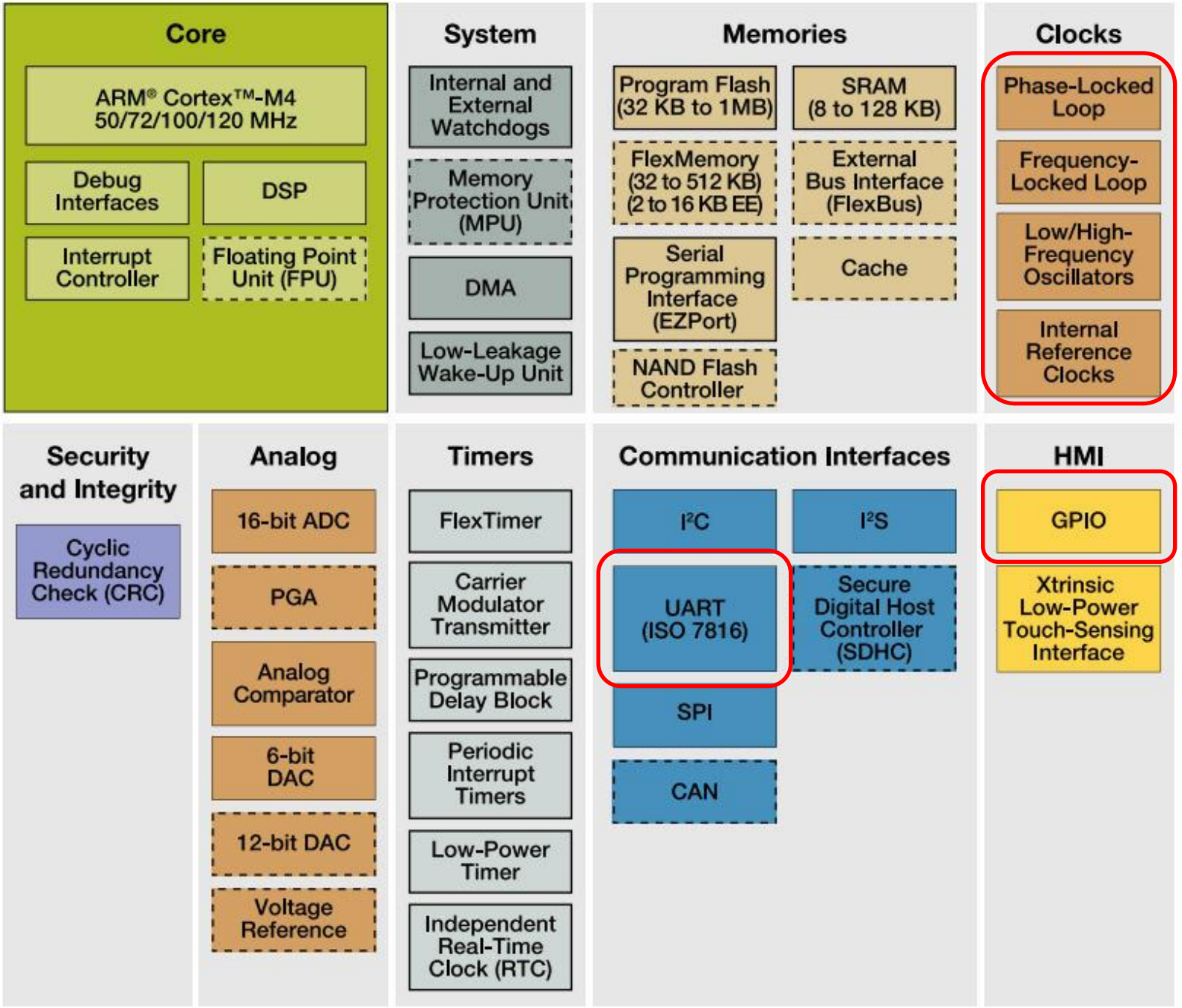
Tsinghua Freescale Embedded System Center
www.tsinghua-freescale.net
by Zeng, Ming zengming@tsinghua.edu.cn
Gong, Guanghua ggh@tsinghua.edu.cn
Xue, Tao gbe.tao.xue@gmail.com

Freescale Kinetis K10 GPIO IRQ Step by Step

清华 Freescale 应用开发研究中心 薛涛

2012年3月


Core
ARM® Cortex™-M4 50/72/100/120 MHz
Debug Interfaces
DSP
Interrupt Controller
Floating Point Unit (FPU)
System
Internal and External Watchdogs
Memory Protection Unit (MPU)
DMA
Low-Leakage Wake-Up Unit
Memories
Program Flash (32 KB to 1MB)
SRAM (8 to 128 KB)
FlexMemory (32 to 512 KB) (2 to 16 KB EE)
External Bus Interface (FlexBus)
Serial Programming Interface (EZPort)
Cache
NAND Flash Controller
Clocks
Phase-Locked Loop
Frequency-Locked Loop
Low/High-Frequency Oscillators
Internal Reference Clocks
Security and Integrity
Cyclic Redundancy Check (CRC)
Analog
16-bit ADC
PGA
Analog Comparator
6-bit DAC
12-bit DAC
Voltage Reference
Timers
FlexTimer
Carrier Modulator Transmitter
Programmable Delay Block
Periodic Interrupt Timers
Low-Power Timer
Independent Real-Time Clock (RTC)
Communication Interfaces
I²C
I²S
UART (ISO 7816)
Secure Digital Host Controller (SDHC)
SPI
CAN
HMI
GPIO
Xtrinsic Low-Power Touch-Sensing Interface
Standard Feature
Optional Feature

FESC

What we used?

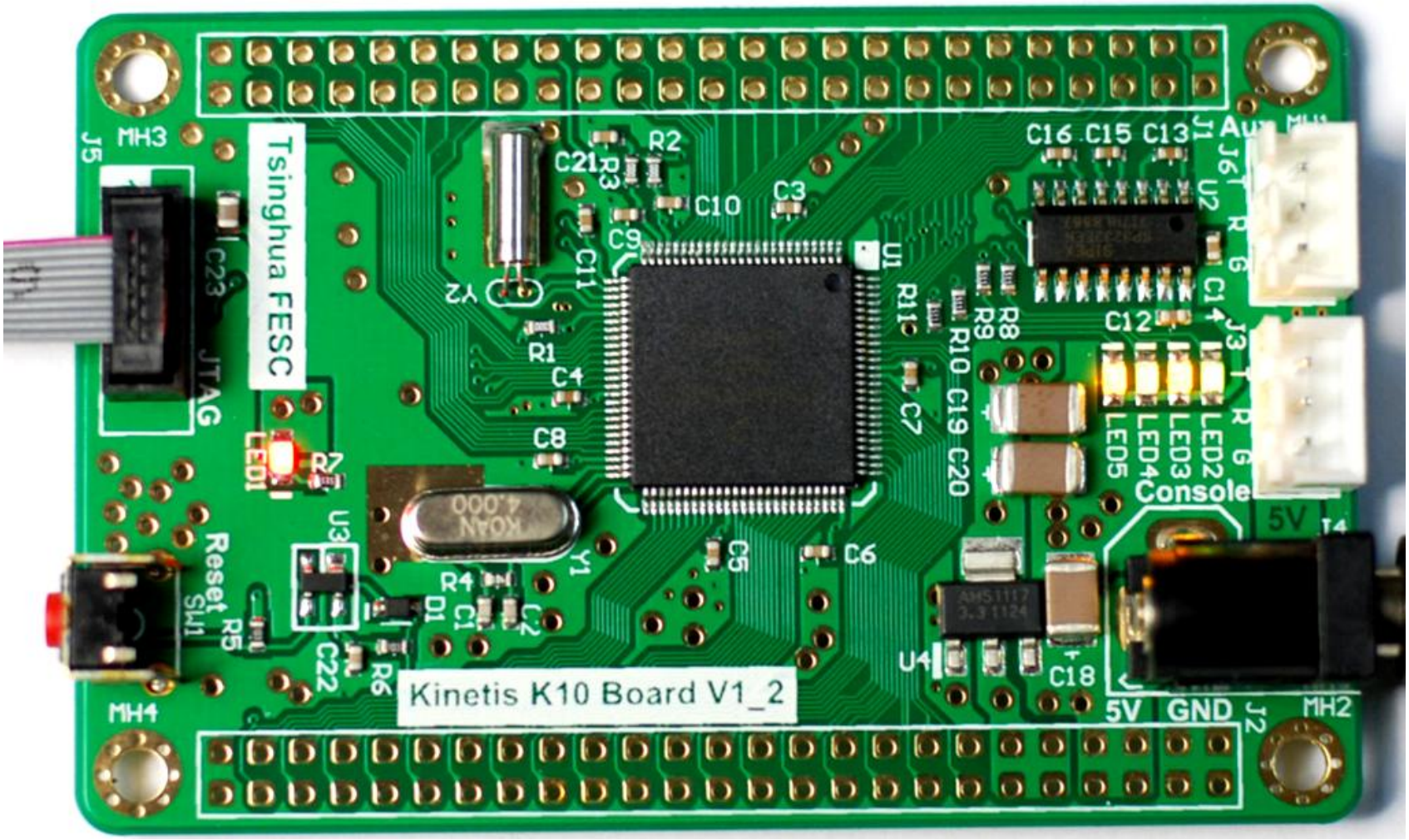
Freescale part number	CPU frequency	Pin count	Package	Total flash memory	Program flash	EEPROM	SRAM	GPIO
MK10DN512ZVLL10	100 MHz	100	LQFP	512 KB	512 KB	—	128 KB	70

Reference :
DDI0403D_arm_architecture_v7m_reference_manual_errata_markup_1_0.pdf
DDI0439B_cortex_m4_r0p0_trm.pdf
K10P100M100SF2RM.pdf
KQRUG.pdf
K10P100M100SF2.pdf
AN4445.pdf

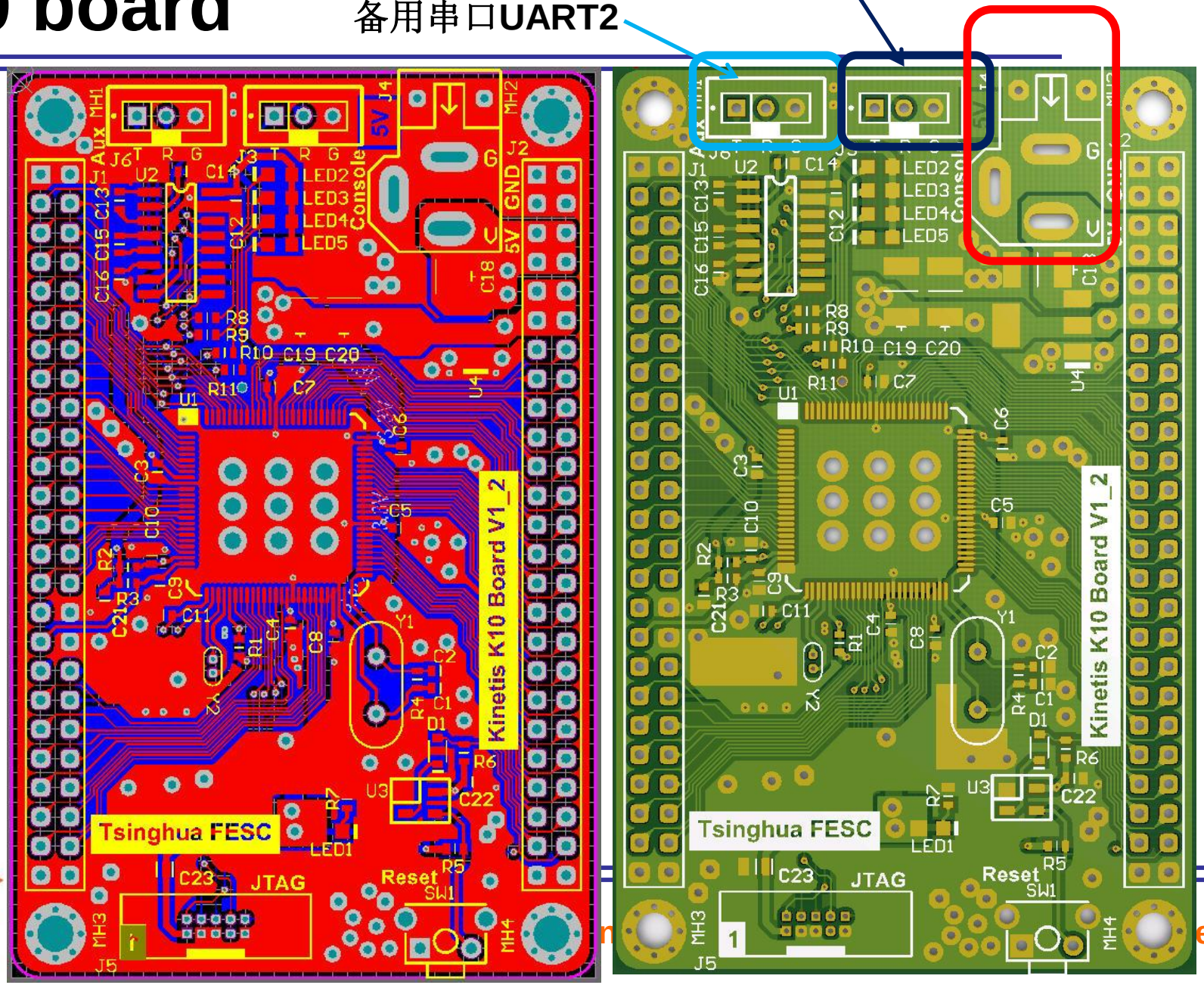
Revision:

Revision	Mask Set	Part Number Example
1.0	0M33Z	PK10N512VMD100
1.1	0N30D	N/A
1.2	1N30D / 2N30D (functionally identical)	PK10N512VMD100
1.4	4N30D	MK10DN512ZVMD10 ('Z' character: INITIAL Production mask set)
2.2	2N22D	MK10DN512VMD10 (no 'Z' character: PRODUCTION mask set)

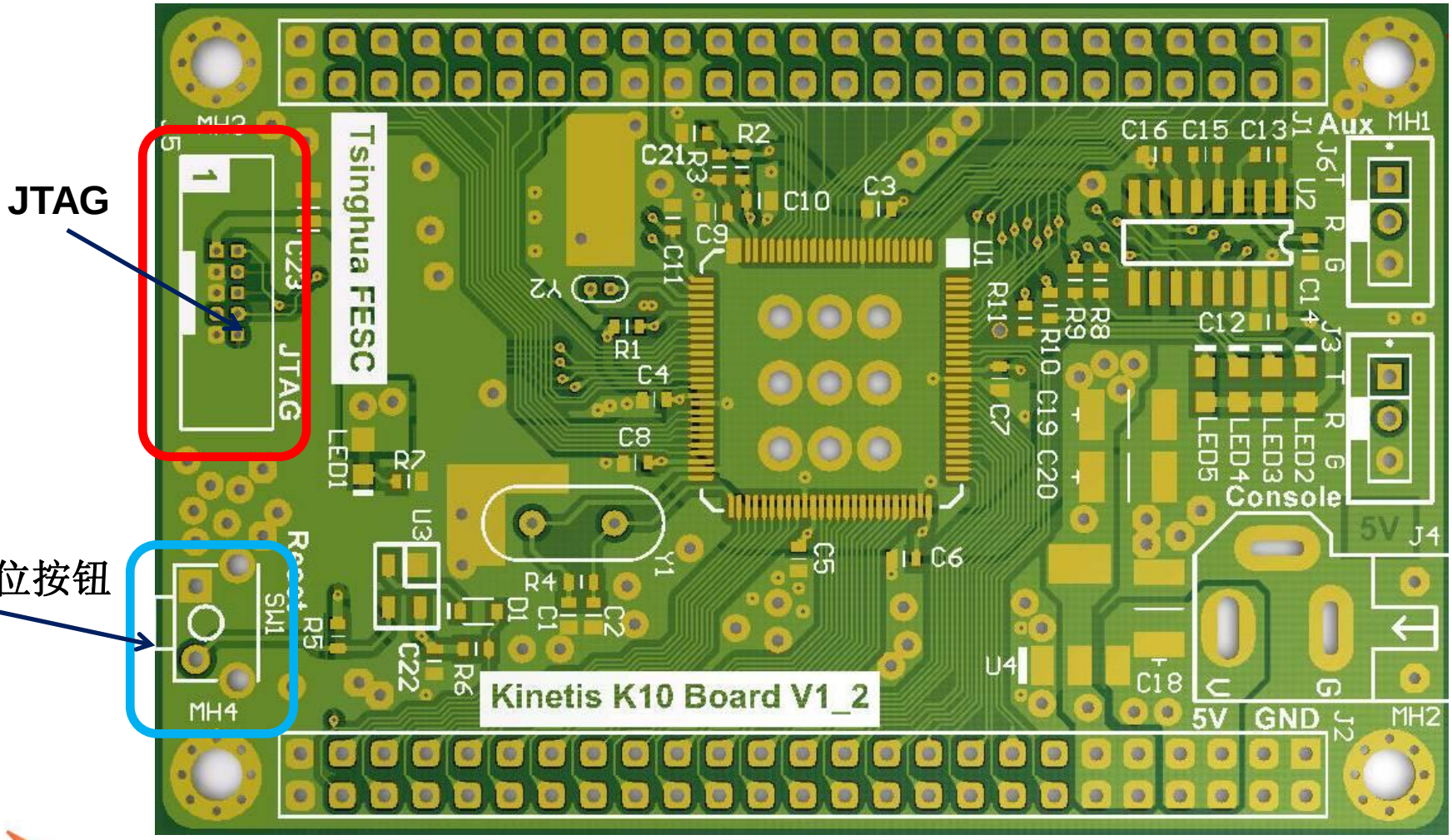
K10 board


Tsinghua FESC
JTAG
Reset
SW1
LED1
Console
5V
GND
Kinetis K10 Board V1_2
MH1
MH2
MH3
MH4
J1
J2
J3
J4
J5
J6
U1
U2
U3
Y1
Y2

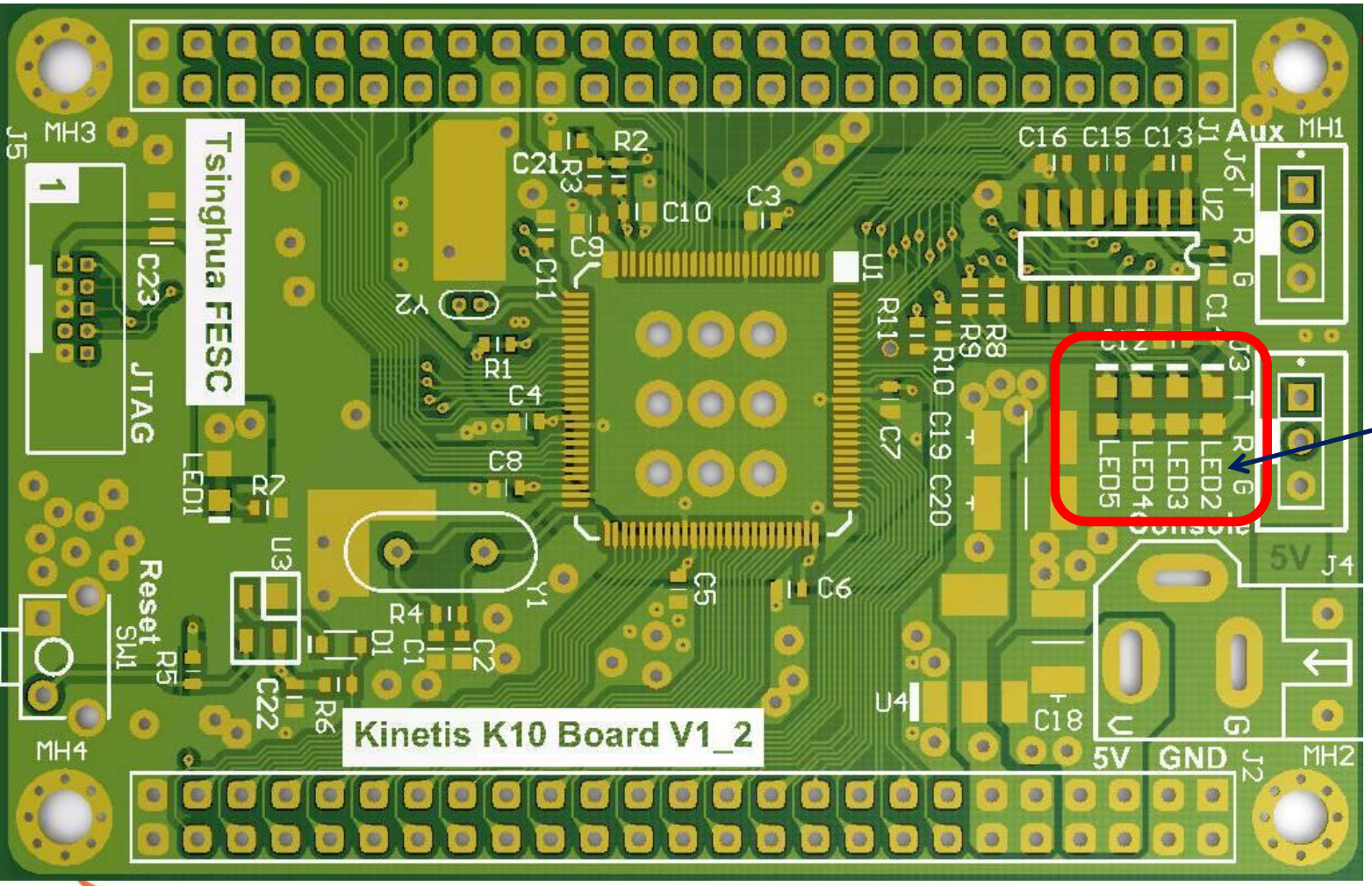
K10 board


调试用串口UART0
电源 直流5V 100mA
备用串口UART2
Kinetis K10 Board V1_2
Tsinghua FESC
JTAG
Reset
LED1
LED2
LED3
LED4
LED5
Console
5V
GND

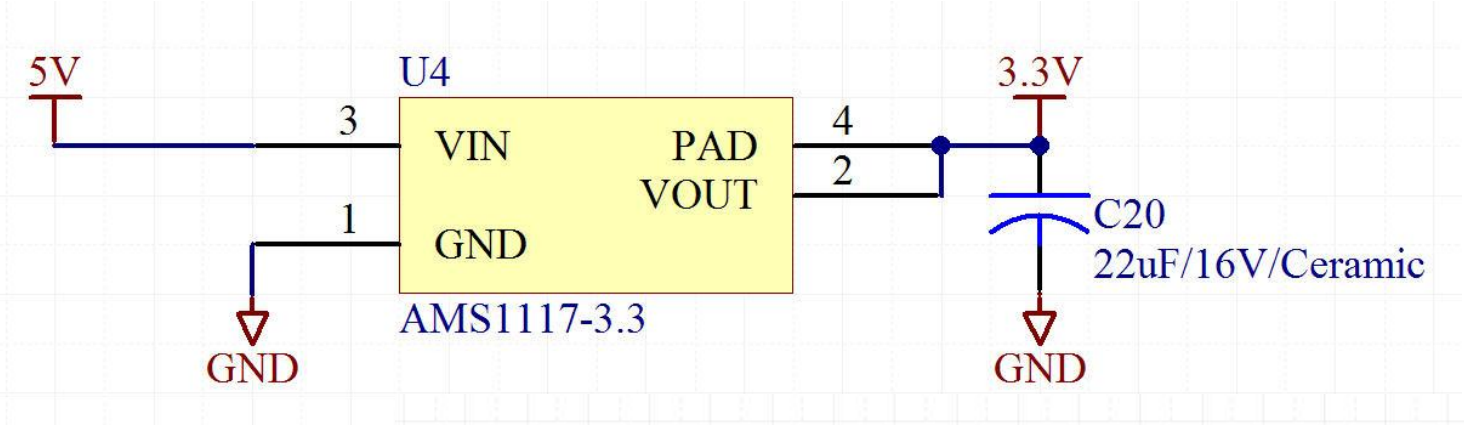
K10 board Debug & Reset button

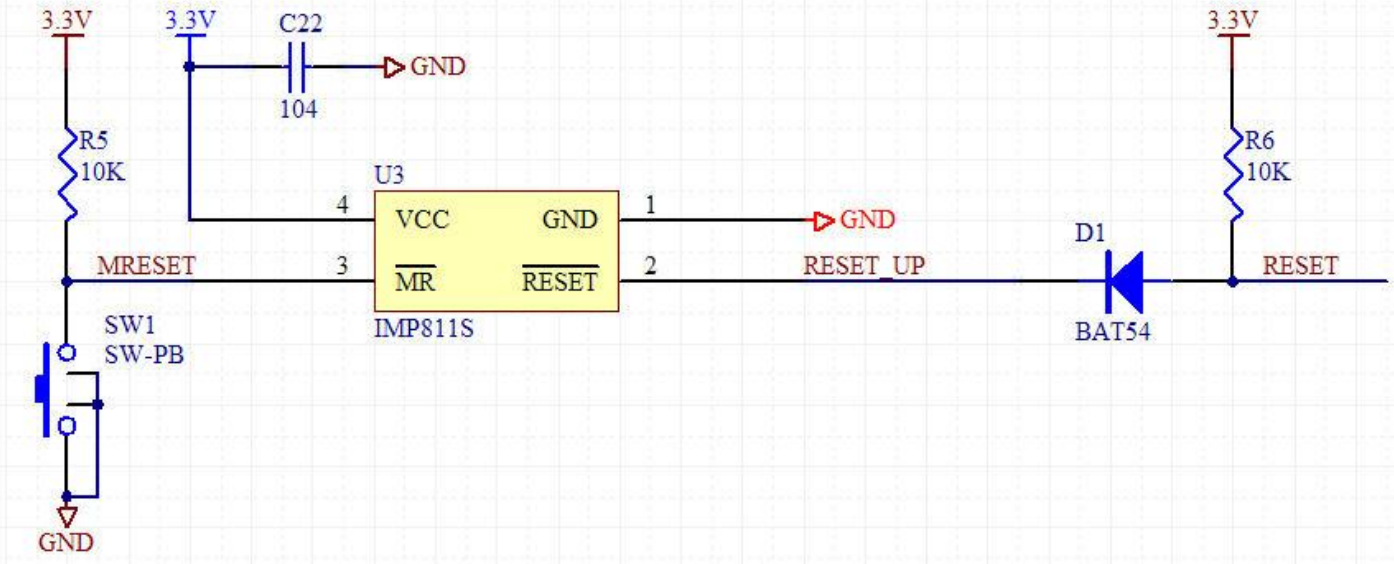


K10 board Leds



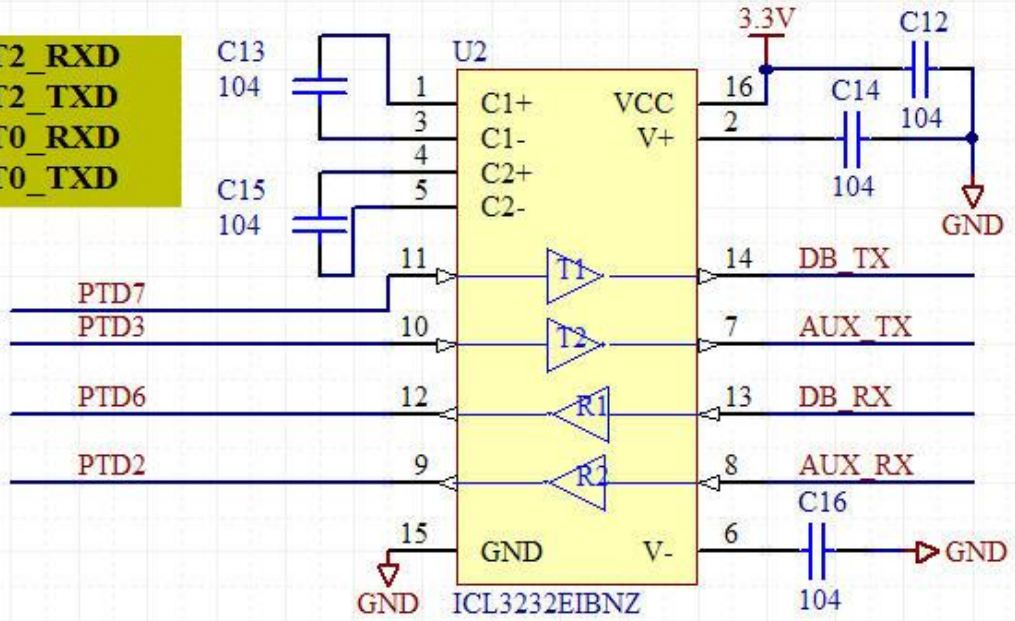
K10 board Power & Reset

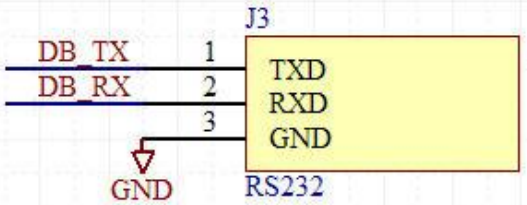


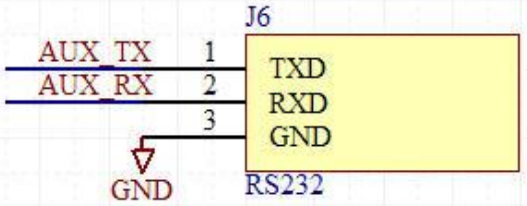


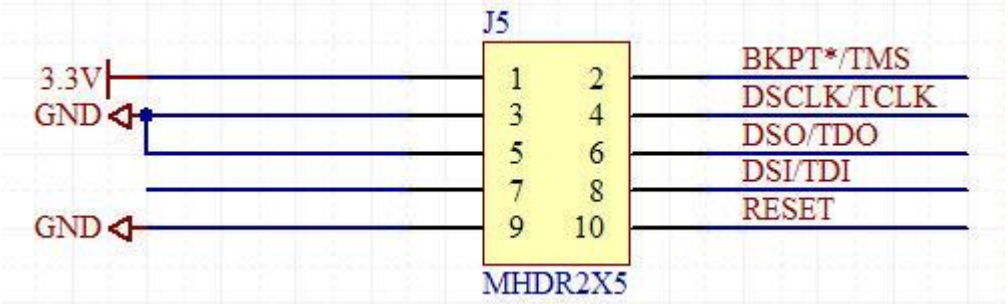
K10 board RS232 & JTAG

PTD2 as UART2_RXD
PTD3 as UART2_TXD
PTD6 as UART0_RXD
PTD7 as UART0_TXD

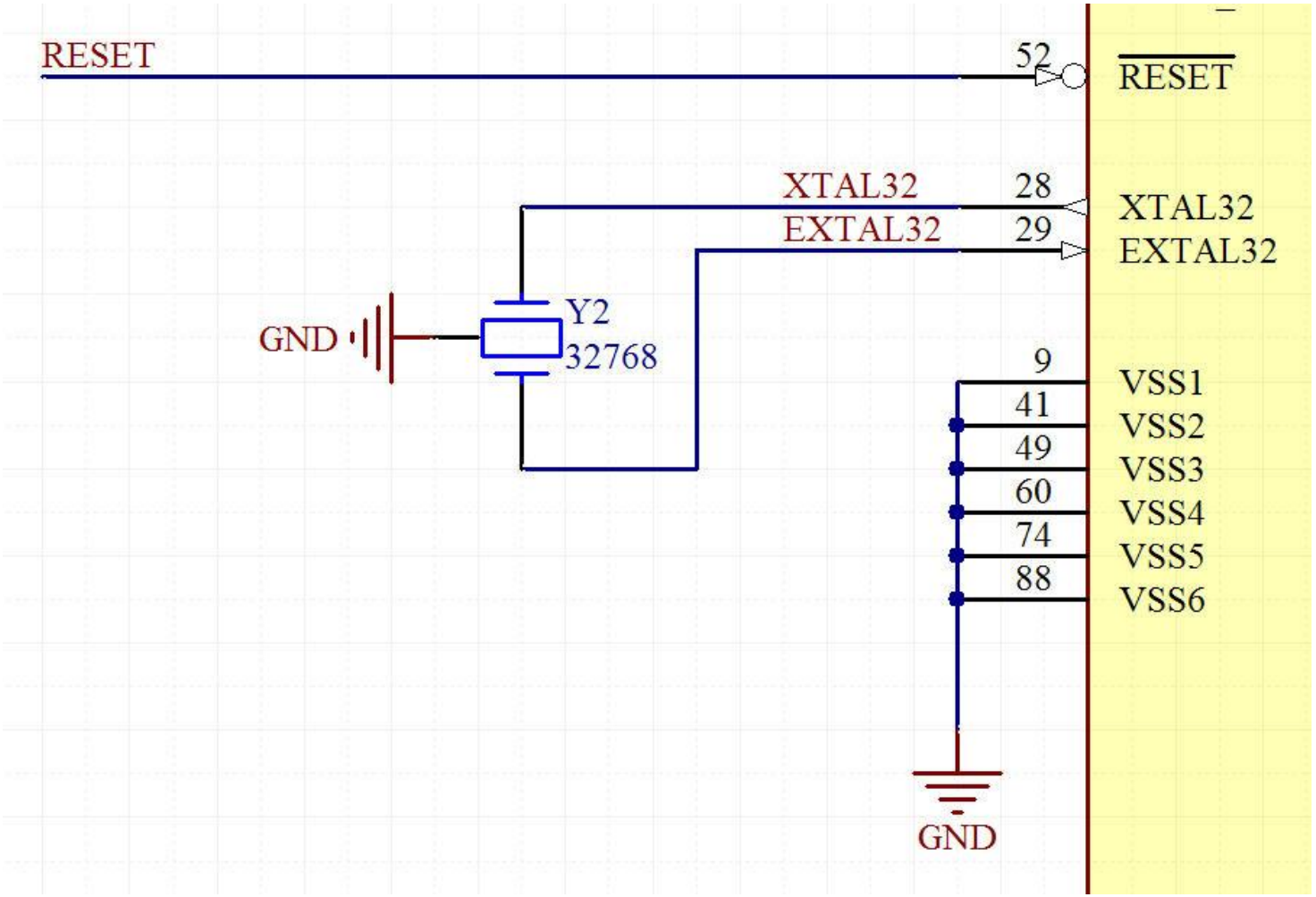




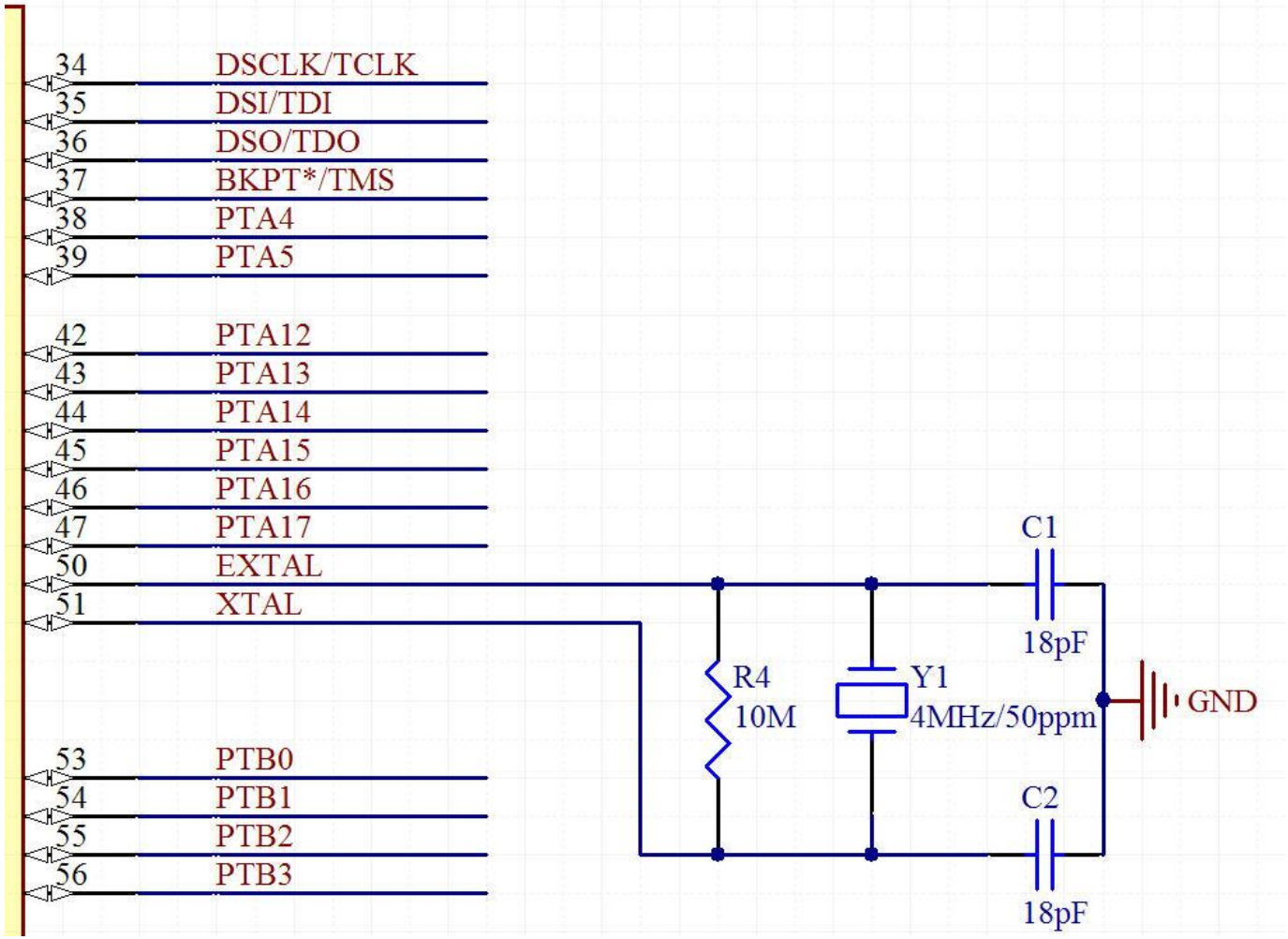




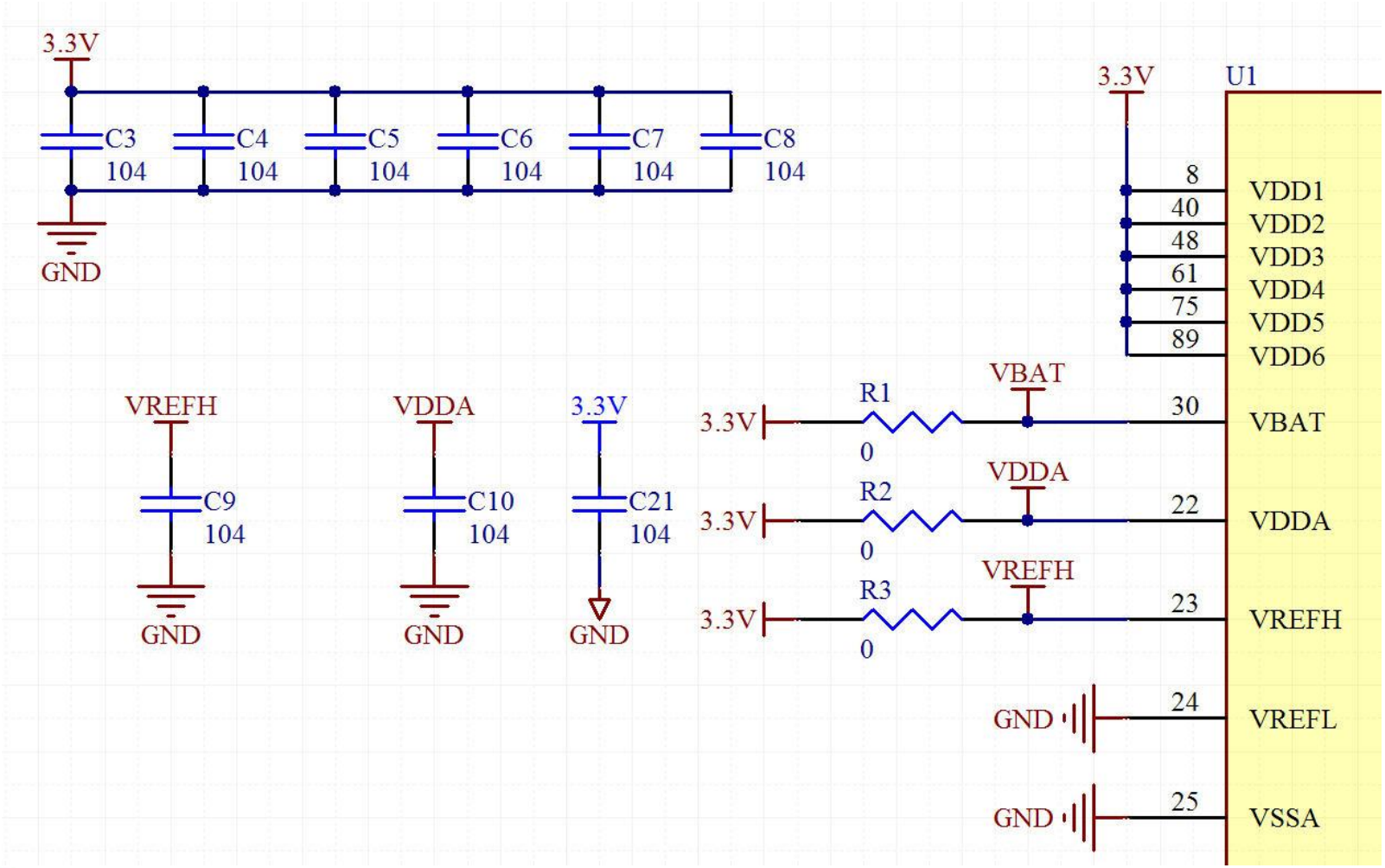
K10 board CLOCK 32768



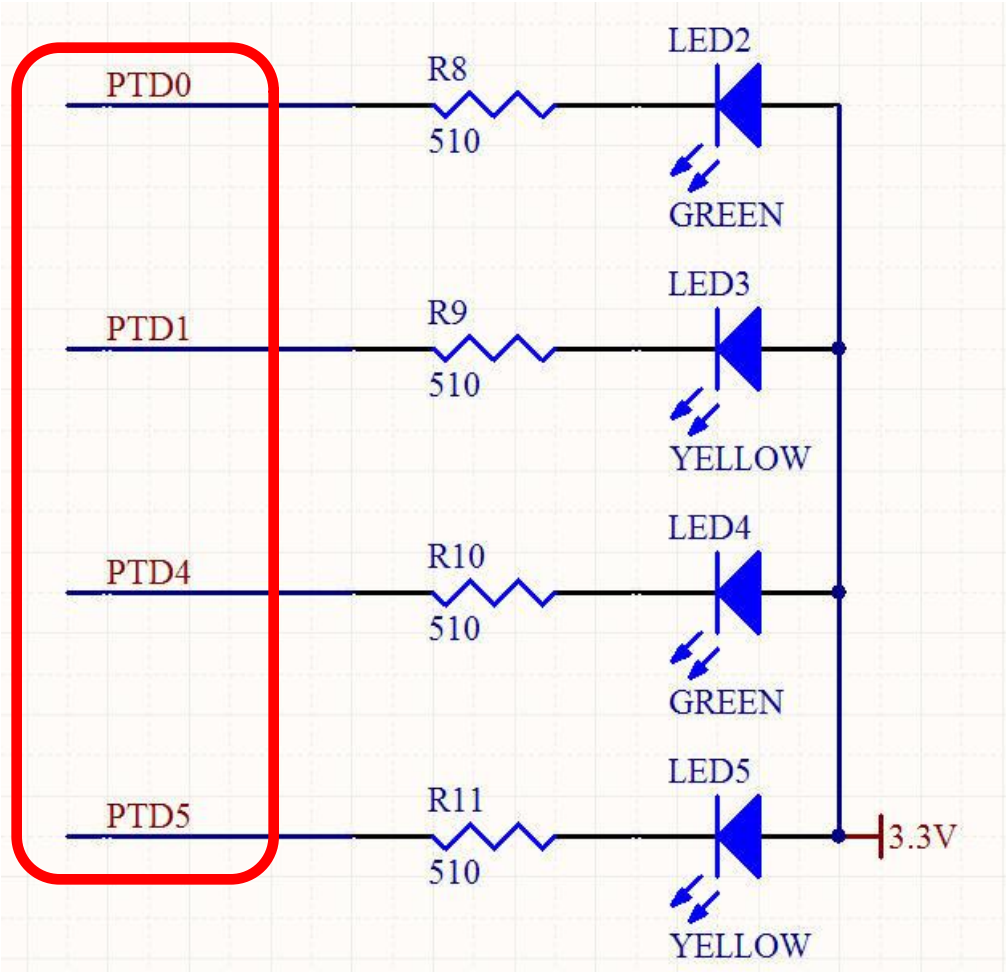
K10 board CLOCK 4MHz



K10 board Power Supply



K10 LEDs


PTD0
PTD1
PTD4
PTD5
R8
R9
R10
R11
510
510
510
510
LED2
LED3
LED4
LED5
GREEN
YELLOW
GREEN
YELLOW
3.3V

K10 GPIO IRQ step by step

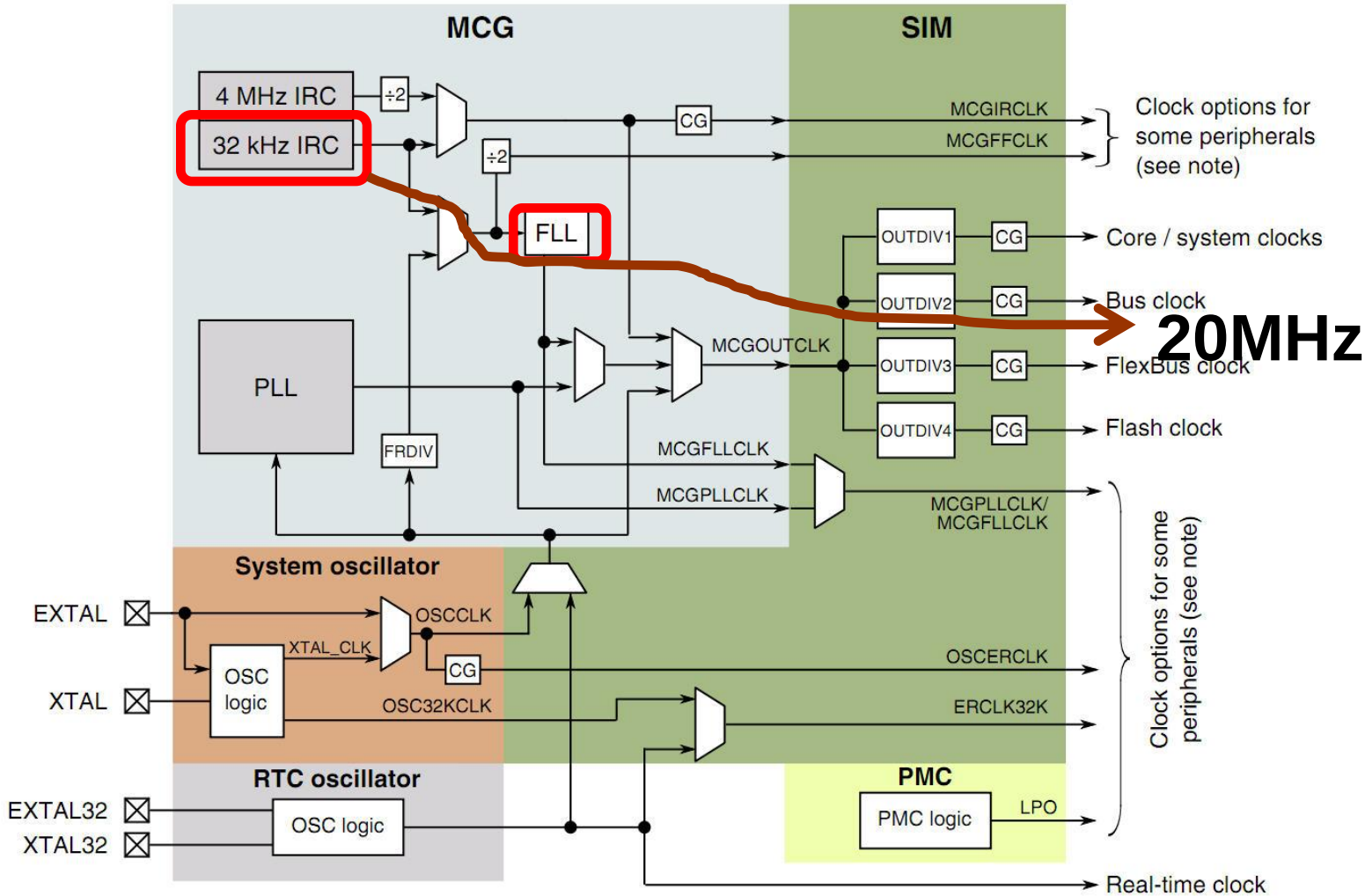
Step 1: SIM enable correspond GPIO port clock

Step 2: PCR enable correspond GPIO port ALT1 function as GPIO IRQ function, enable IRQ

Step 3: Write the ISR

Step 4: Send a key to generate IRQ through RS232 port

K10 Clock previous MCG/SIM/OSC



K10 SIM

SIM memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_7000	System Options Register 1 (SIM_SOPT1)	32	R/W	Undefined	12.2.1/250
4004_8004	System Options Register 2 (SIM_SOPT2)	32	R/W	0000_1000h	12.2.2/252
4004_800C	System Options Register 4 (SIM_SOPT4)	32	R/W	0000_0000h	12.2.3/254
4004_8010	System Options Register 5 (SIM_SOPT5)	32	R/W	0000_0000h	12.2.4/257
4004_8014	System Options Register 6 (SIM_SOPT6)	32	R/W	0000_0000h	12.2.5/258
4004_8018	System Options Register 7 (SIM_SOPT7)	32	R/W	0000_0000h	12.2.6/259
4004_8024	System Device Identification Register (SIM_SDID)	32	R	Undefined	12.2.7/261
4004_8028	System Clock Gating Control Register 1 (SIM_SCGC1)	32	R/W	0000_0000h	12.2.8/262
4004_802C	System Clock Gating Control Register 2 (SIM_SCGC2)	32	R/W	0000_0000h	12.2.9/263
4004_8030	System Clock Gating Control Register 3 (SIM_SCGC3)	32	R/W	0000_0000h	12.2.10/ 264
4004_8034	System Clock Gating Control Register 4 (SIM_SCGC4)	32	R/W	6010_0030h	12.2.11/ 265
4004_8038	System Clock Gating Control Register 5 (SIM_SCGC5)	32	R/W	0004_0180h	12.2.12/ 267
4004_803C	System Clock Gating Control Register 6 (SIM_SCGC6)	32	R/W	4000_0001h	12.2.13/ 269
4004_8040	System Clock Gating Control Register 7 (SIM_SCGC7)	32	R/W	0000_0007h	12.2.14/ 272
4004_8044	System Clock Divider Register 1 (SIM_CLKDIV1)	32	R/W	Undefined	12.2.15/ 273
4004_8048	System Clock Divider Register 2 (SIM_CLKDIV2)	32	R/W	0000_0000h	12.2.16/ 275
4004_804C	Flash Configuration Register 1 (SIM_FCFG1)	32	R	Undefined	12.2.17/ 276
4004_8050	Flash Configuration Register 2 (SIM_FCFG2)	32	R	Undefined	12.2.18/ 277
4004_8054	Unique Identification Register High (SIM_UIDH)	32	R	Undefined	12.2.19/ 278
4004_8058	Unique Identification Register Mid-High (SIM_UIDMH)	32	R	Undefined	12.2.20/ 279
4004_805C	Unique Identification Register Mid Low (SIM_UIDML)	32	R	Undefined	12.2.21/ 279
4004_8060	Unique Identification Register Low (SIM_UIDL)	32	R	Undefined	12.2.22/ 280

FESC

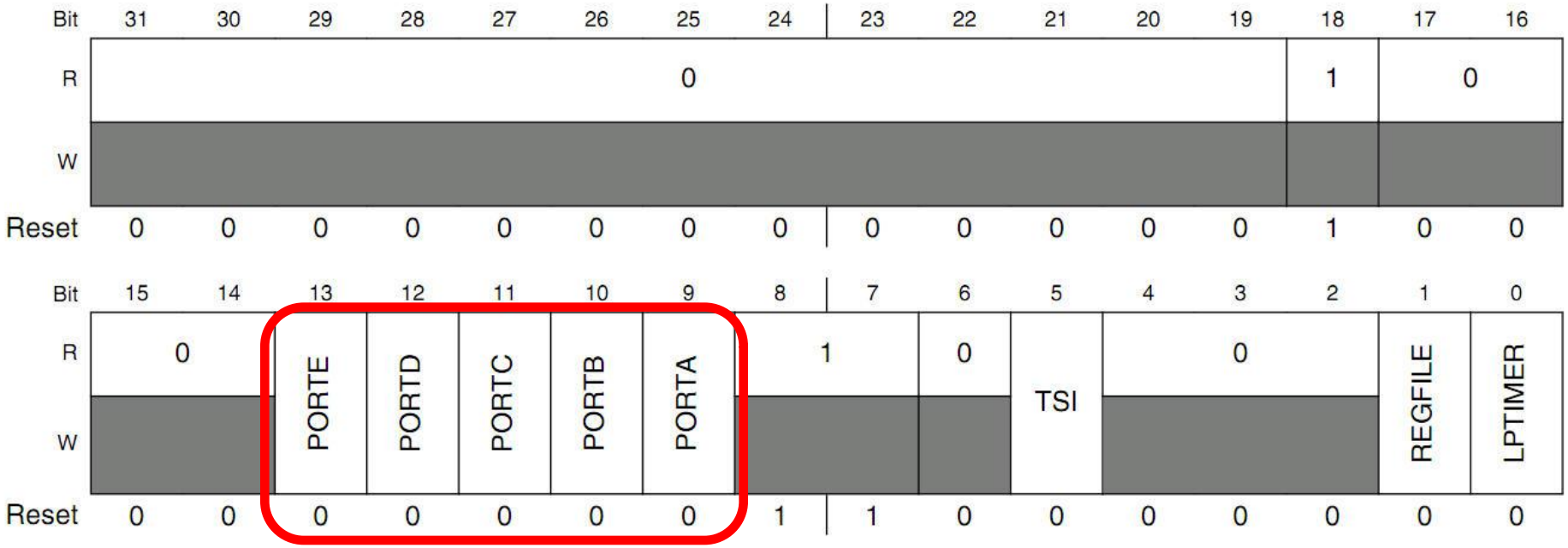
K10 SIM C Code

```
/* Turn on all port clocks */
SIM_SCGC5 |= SIM_SCGC5_PORTD_MASK;
```

K10 SIM

12.2.12 System Clock Gating Control Register 5 (SIM_SCGC5)

Address: SIM_SCGC5 is 4004_7000h base + 1038h offset = 4004_8038h



K10 SIM

13 PORTE	Port E Clock Gate Control This bit controls the clock gate to the Port E module. 0 Clock disabled 1 Clock enabled
12 PORTD	Port D Clock Gate Control This bit controls the clock gate to the Port D module. 0 Clock disabled 1 Clock enabled
11 PORTC	Port C Clock Gate Control This bit controls the clock gate to the Port C module. 0 Clock disabled 1 Clock enabled
10 PORTB	Port B Clock Gate Control This bit controls the clock gate to the Port B module. 0 Clock disabled 1 Clock enabled
9 PORTA	Port A Clock Gate Control This bit controls the clock gate to the Port A module. 0 Clock disabled 1 Clock enabled

K10 GPIO Configure

Control Mode:

PORTx_PCRn	Pin Control Register
PORTx_GPCLR	Global Pin Control Low Register
PORTx_GPCHR	Global Pin Control High Register
PORTx_ISFR	Interrupt Status Flag Register
PORTx_DFER	Digital Filter Enable Register
PORTx_DFCR	Digital Filter Clock Register
PORTx_DFWR	Digital Filter Width Register

K10 GPIO PCR

Detail in Page 239 of K10P100M100SF2RM.pdf

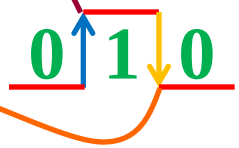
Addresses: 4004_9000h base + 0h offset + (4d × *n*), where *n* = 0d to 31d

Bit	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
R	0							ISF	0				IRQC				LK	0						MUX			0	DSE	ODE	PFE	0	SRE	PE	PS
W								w1c																								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Interrupt Status Flag

Interrupt Configuration

0000 Interrupt/DMA Request disabled.
0001 DMA Request on rising edge.
0010 DMA Request on falling edge.
0011 DMA Request on either edge.
0100 Reserved.
1000 Interrupt when logic zero.
1001 Interrupt on rising edge.
1010 Interrupt on falling edge.
1011 Interrupt on either edge.
1100 Interrupt when logic one.



Drive Strength Enable

Open Drain Enable

Passive Filter Enable

Slow Rate Enable

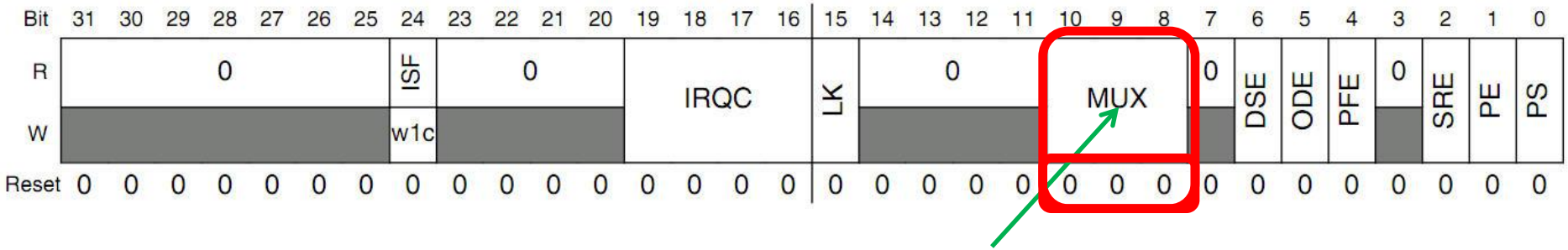
Pull Enable

Pull Select
Pull Up
Pull Down

K10 GPIO PCR

Detail in Page 239 of K10P100M100SF2RM.pdf

Addresses: 4004_9000h base + 0h offset + (4d × *n*), where *n* = 0d to 31d



Pin Mux Control

000 Pin Disabled (Analog).

001 Alternative 1 (GPIO).

010 Alternative 2 (chip specific).

011 Alternative 3 (chip specific).

100 Alternative 4 (chip specific).

101 Alternative 5 (chip specific).

110 Alternative 6 (chip specific).

111 Alternative 7 (chip specific / JTAG / NMI).

K10 Signal Multiplexing and Pin Assignments

Find in Page 59 of K10P100M100SF2.pdf

K10 GPIO PCR

K10 Signal Multiplexing and Pin Assignments
Find in Page 59 of K10P100M100SF2.pdf
Detail in Page 239 of K10P100M100SF2RM.pdf

Addresses: 4004_9000h base + 0h offset + (4d × *n*), where *n* = 0d to 31d

Bit	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
R	0							ISF	0				IRQC				LK	0				MUX			0	DSE	ODE	PFE	0	SRE	PE	PS
W								w1c																								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

100 LQFP	Pin Name	Default	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5	ALT6	ALT7	EzPort
								FB_BE15_8_BLS23_16_b			
93	PTD0			PTD0	SPI0_PCS0	UART2_RTS_b		FB_ALE/FB_CS1_b/FB_TS_b			
94	PTD1	/ADC0_SE5b	/ADC0_SE5b	PTD1	SPI0_SCK	UART2_CTS_b		FB_CS0_b			
95	PTD2			PTD2	SPI0_SOUT	UART2_RX		FB_AD4			
96	PTD3			PTD3	SPI0_SIN	UART2_TX		FB_AD3			
97	PTD4			PTD4	SPI0_PCS1	UART0_RTS_b	FTM0_CH4	FB_AD2	EWM_IN		
98	PTD5	/ADC0_SE6b	/ADC0_SE6b	PTD5	SPI0_PCS2	UART0_CTS_b	FTM0_CH5	FB_AD1	EWM_OUT_b		
99	PTD6	/ADC0_SE7b	/ADC0_SE7b	PTD6	SPI0_PCS3	UART0_RX	FTM0_CH6	FB_AD0	FTM0_FLT0		
100	PTD7			PTD7	CMT_IRO	UART0_TX	FTM0_CH7		FTM0_FLT1		

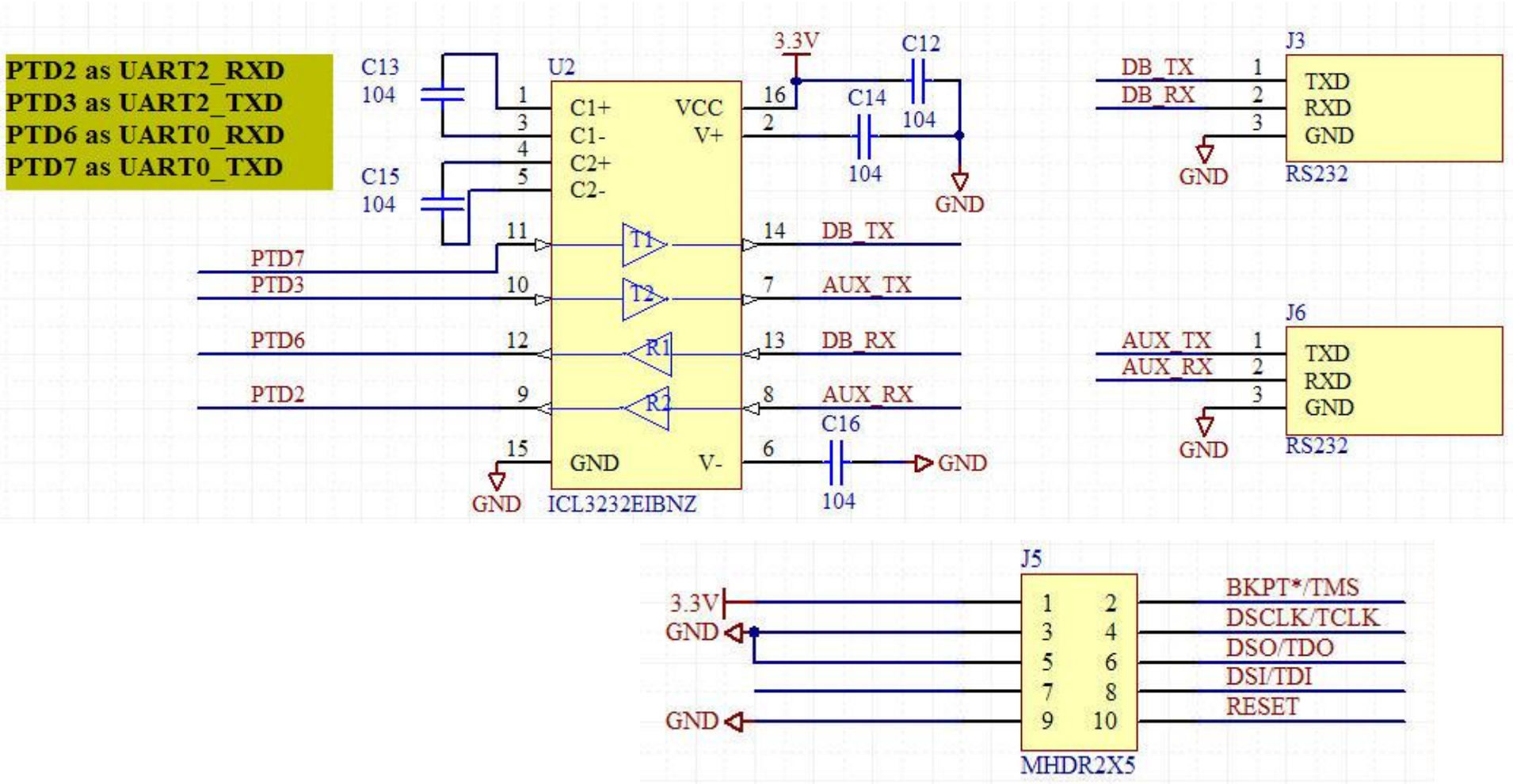
K10 GPIO PCR C Code

```
/* Set pin2 of PORTD as GPIO IRQ function */
PORTD_PCR2 = PORT_PCR_MUX(1)|
PORT_PCR_IRQC(0xA)|
PORT_PCR_PE_MASK|
PORT_PCR_PS_MASK;
```

K10 GPIO PCR C Code

```
/* Set pin3 of PORTD as GPIO output and high level */
PORTD_PCR3 = (0|PORT_PCR_MUX(1));
GPIOD_PDDR=GPIO_PDDR_PDD(GPIO_PIN(3));
GPIOD_PDOR |= GPIO_PDOR_PDO(GPIO_PIN(3));
```

K10 board RS232 & JTAG



K10 GPIC

NVIC

Detail in Page 65 of K10P100M100SF2RM.pdf

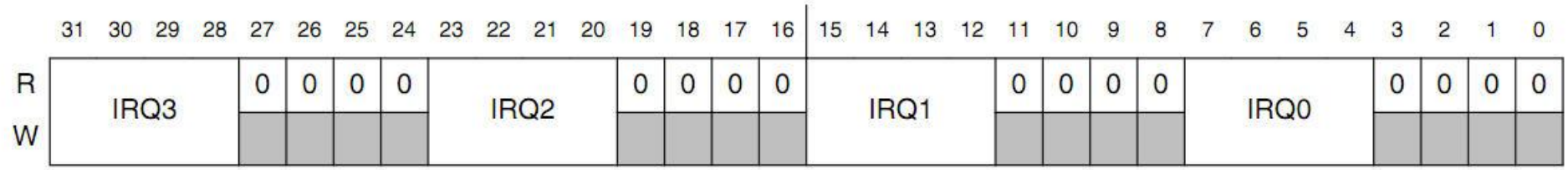
Address	Vector	IRQ[1]	NVIC non-IPR register number [2]	NVIC IPR register number [3]	Source module	Source description
ARM Core System Handler Vectors						
0x0000_0000	0	–	–	–	ARM core	Initial Stack Pointer
0x0000_0004	1	–	–	–	ARM core	Initial Program Counter
0x0000_0008	2	–	–	–	ARM core	Non-maskable Interrupt (NMI)
0x0000_000C	3	–	–	–	ARM core	Hard Fault
0x0000_0010	4	–	–	–	ARM core	MemManage Fault
0x0000_0014	5	–	–	–	ARM core	Bus Fault
0x0000_0018	6	–	–	–	ARM core	Usage Fault
0x0000_001C	7	–	–	–	—	—
0x0000_0020	8	–	–	–	—	—
0x0000_0024	9	–	–	–	—	—
0x0000_0028	10	–	–	–	—	—
0x0000_002C	11	–	–	–	ARM core	Supervisor call (SVCall)
0x0000_0030	12	–	–	–	ARM core	Debug Monitor
0x0000_0034	13	–	–	–	—	—
0x0000_0038	14	–	–	–	ARM core	Pendable request for system service (PendableSrvReq)
0x0000_003C	15	–	–	–	ARM core	System tick timer (SysTick)
Non-Core Vectors						
0x0000_0040	16	0	0	0	DMA	DMA channel 0 transfer complete
0x0000_0044	17	1	0	0	DMA	DMA channel 1 transfer complete
0x0000_0048	18	2	0	0	DMA	DMA channel 2 transfer complete
0x0000_004C	19	3	0	0	DMA	DMA channel 3 transfer complete
0x0000_0050	20	4	0	1	DMA	DMA channel 4 transfer complete

Table continues on the next page...

FESC

K10 GPIO IRQ

Address	Vector	IRQ[1]	NVIC non-IPR register number [2]	NVIC IPR register number [3]	Source module	Source description
0x0000_01A4	105	89	2	22	Port control module	Pin detect (Port C)
0x0000_01A8	106	90	2	22	Port control module	Pin detect (Port D)
0x0000_01AC	107	91	2	22	Port control module	Pin detect (Port E)
0x0000_01B0	108	92	2	23	—	—
0x0000_01B4	109	93	2	23	—	—
0x0000_01B8	110	94	2	23	Software	Software interrupt[4]



K10 NVIC

Detail in Page 69 of DDI0439B_cortex_m4_r0p0_trm.pdf

Address	Name	Type	Reset	Description
0xE000E004	ICTR	RO	-	*Interrupt Controller Type Register, ICTR*
0xE000E100 - 0xE000E11C	NVIC_ISER0 - NVIC_ISER7	RW	0x00000000	Interrupt Set-Enable Registers
0xE000E180 - 0E000xE19C	NVIC_ICER0 - NVIC_ICER7	RW	0x00000000	Interrupt Clear-Enable Registers
0xE000E200 - 0xE000E21C	NVIC_ISPR0 - NVIC_ISPR7	RW	0x00000000	Interrupt Set-Pending Registers
0xE000E280 - 0xE000E29C	NVIC_ICPR0 - NVIC_ICPR7	RW	0x00000000	Interrupt Clear-Pending Registers
0xE000E300 - 0xE000E31C	NVIC_IABR0 - NVIC_IABR7	RO	0x00000000	Interrupt Active Bit Register
0xE000E400 - 0xE000E41F	NVIC_IPR0 - NVIC_IPR59	RW	0x00000000	Interrupt Priority Register

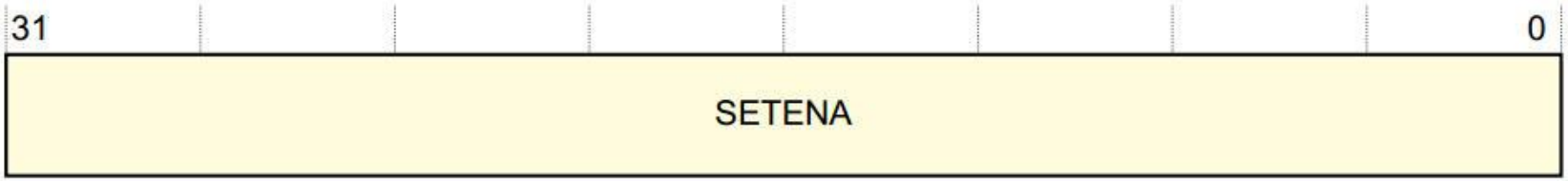
K10 NVIC

Detail in Page 755 of DDI0403D_arm_architecture_v7m_reference_manual_errata_markup_1_0.pdf

The NVIC_ISER characteristics are:

Purpose	Enables, or reads the enable state of a group of interrupts.
Usage constraints	NVIC_ISER*n*[31:0] are the set-enable bits for interrupts (31+(32***n*)) - (32***n*). When *n*=15, bits [31:16] are reserved.
Configurations	At least one register is always implemented, see *Implemented NVIC registers* on page B3-753.
Attributes	See Table B3-35 on page B3-753.

Figure B3-28 shows the NVIC_ISER*n* bit assignments.

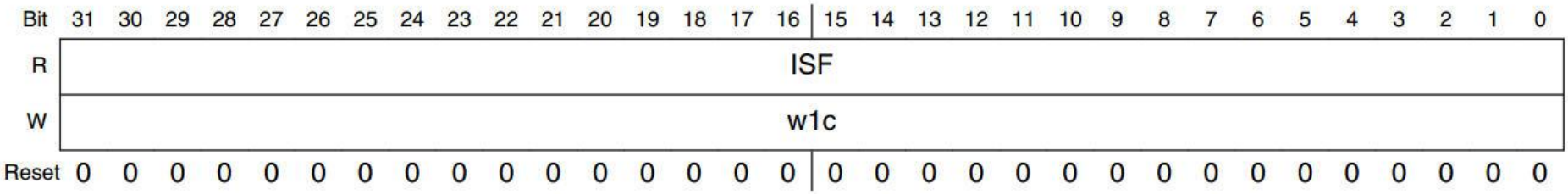


K10 NVIC C Code

```
void enable_irq (int irq)
{
    int div;
    /* Determine which of the NVICISERs corresponds to the irq */
    div = irq/32;
    switch (div)
    {
    case 0x0:
            NVICICPR0 = 1 << (irq%32);
            NVICISER0 = 1 << (irq%32);
            break;
    case 0x1:
            NVICICPR1 = 1 << (irq%32);
            NVICISER1 = 1 << (irq%32);
            break;
    case 0x2:
            NVICICPR2 = 1 << (irq%32);
            NVICISER2 = 1 << (irq%32);
            break;
    }
}
```

K10 GPIO PCR

Detail in Page 241 of K10P100M100SF2RM.pdf



Bit	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
R	ISF																															
W	w1c																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Write 1 to the correspond bit clear the IRQ flag

```
PORTD_ISFR |= 0x00000004;  //Clear Port D pin 2 ISR flags
```

K10 GPIO ISR C Code

Detail in kinetis_sysinit.c

```
#pragma define_section vectortable ".vectortable" ".vectortable" ".vectortable" far_abs R
static __declspec(vectortable) tVectorTable __vect_table = { /* Interrupt vector table */
  __SP_INIT,                                    /* 0 (0x00000000) (prior: -) */
 {
  (tIsrFunc)__thumb_startup,                    /* 1 (0x00000004) (prior: -) */
  (tIsrFunc)isrINT_NMI,                         /* 2 (0x00000008) (prior: -2) */
  (tIsrFunc)UNASSIGNED_ISR,                     /* 3 (0x0000000C) (prior: -1) */

  (tIsrFunc)UNASSIGNED_ISR,                     /* 103 (0x0000019C) (prior: -) */
  (tIsrFunc)UNASSIGNED_ISR,                     /* 104 (0x000001A0) (prior: -) */
  (tIsrFunc)UNASSIGNED_ISR,                     /* 105 (0x000001A4) (prior: -) */
  (tIsrFunc)isr_portd_pin2,                     /* 106 (0x000001A8) (prior: -) */
  (tIsrFunc)UNASSIGNED_ISR,                     /* 107 (0x000001AC) (prior: -) */
  (tIsrFunc)UNASSIGNED_ISR,                     /* 108 (0x000001B0) (prior: -) */
  (tIsrFunc)UNASSIGNED_ISR,                     /* 109 (0x000001B4) (prior: -) */
  (tIsrFunc)UNASSIGNED_ISR,                     /* 110 (0x000001B8) (prior: -) */
  (tIsrFunc)UNASSIGNED_ISR,                     /* 111 (0x000001BC) (prior: -) */
  (tIsrFunc)UNASSIGNED_ISR,                     /* 112 (0x000001C0) (prior: -) */
  (tIsrFunc)UNASSIGNED_ISR,                     /* 113 (0x000001C4) (prior: -) */
  (tIsrFunc)UNASSIGNED_ISR,                     /* 114 (0x000001C8) (prior: -) */
  (tIsrFunc)UNASSIGNED_ISR,                     /* 115 (0x000001CC) (prior: -) */
  (tIsrFunc)UNASSIGNED_ISR,                     /* 116 (0x000001D0) (prior: -) */
  (tIsrFunc)UNASSIGNED_ISR,                     /* 117 (0x000001D4) (prior: -) */
  (tIsrFunc)UNASSIGNED_ISR,                     /* 118 (0x000001D8) (prior: -) */
  (tIsrFunc)UNASSIGNED_ISR                      /* 119 (0x000001DC) (prior: -) */
 }
```

```
void isr_portd_pin2(void)
{
        PORTD_ISFR |= 0x00000004;  //Clear Port D pin 2 ISR flags
        counter++;
}
```

K10 GPIO IRQ C Code

```
int main(void)
{
/* Turn on all port clocks */
SIM_SCGC5 |= SIM_SCGC5_PORTD_MASK;
/* Set pin2 of PORTD as GPIO IRQ function */
PORTD_PCR2 =
PORT_PCR_MUX(1)|PORT_PCR_IRQC(0xA)|PORT_PCR_PE_MASK|PORT_PCR_PS_MASK;
/* Set pin3 as output high level */
PORTD_PCR3 = (0|PORT_PCR_MUX(1));
GPIOD_PDDR=GPIO_PDDR_PDD(GPIO_PIN(3));
GPIOD_PDOR |= GPIO_PDOR_PDO(GPIO_PIN(3));
enable_irq(90); //GPIOD Vector is 106. IRQ# is 106-16=90
for(;;)
{
   asm("nop");
   asm("nop");
}
return 0;
}
```

K10 GPIO IRQ

Send a key of SPACE(0x20) to Aux UART of K10 board,

how many GPIO IRQs will you get every SPACE ?