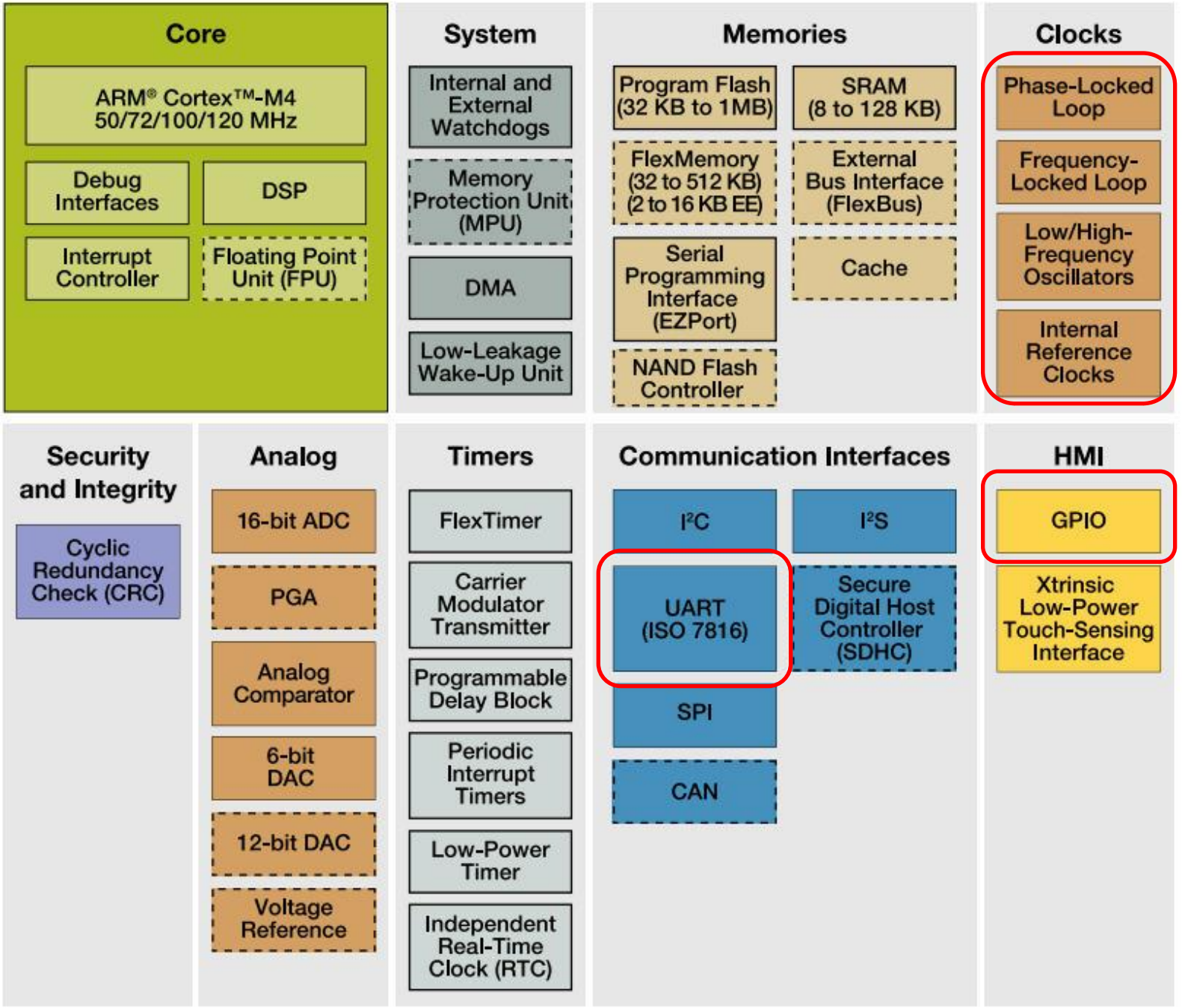
Tsinghua Freescale Embedded System Center
www.tsinghua-freescale.net
by Zeng, Ming zengming@tsinghua.edu.cn
Gong, Guanghua ggh@tsinghua.edu.cn
Xue, Tao gbe.tao.xue@gmail.com

Freescale Kinetis K10 UART 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?

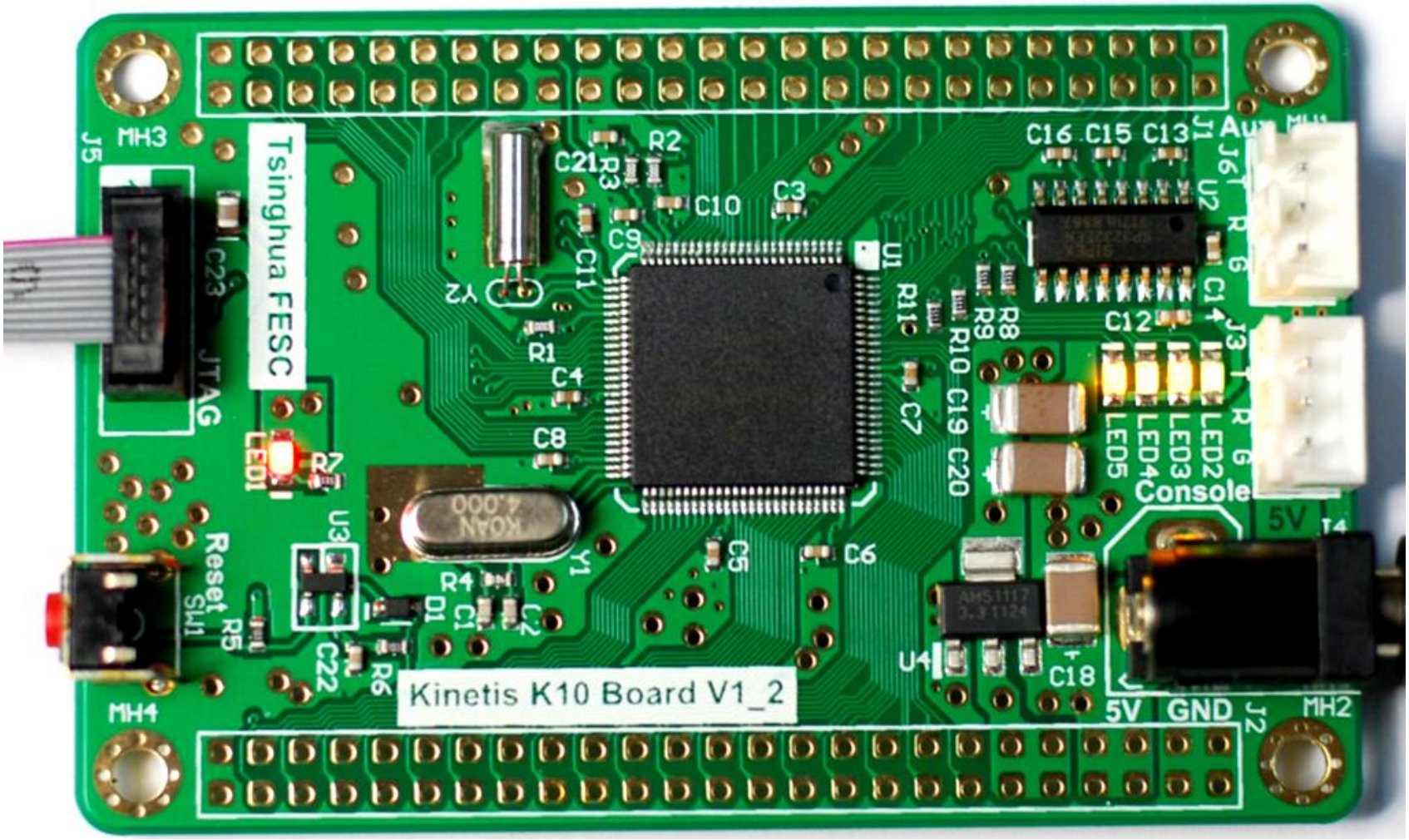
Freescale part number	CPU frequency	Pin count	Package	Total flash memory	Program flash	EEPROM	SRAM	GPIO
MK10DN512ZVLL10	100 MHz	100	LQFP	512 KB	512 KB	—	128 KB	70

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

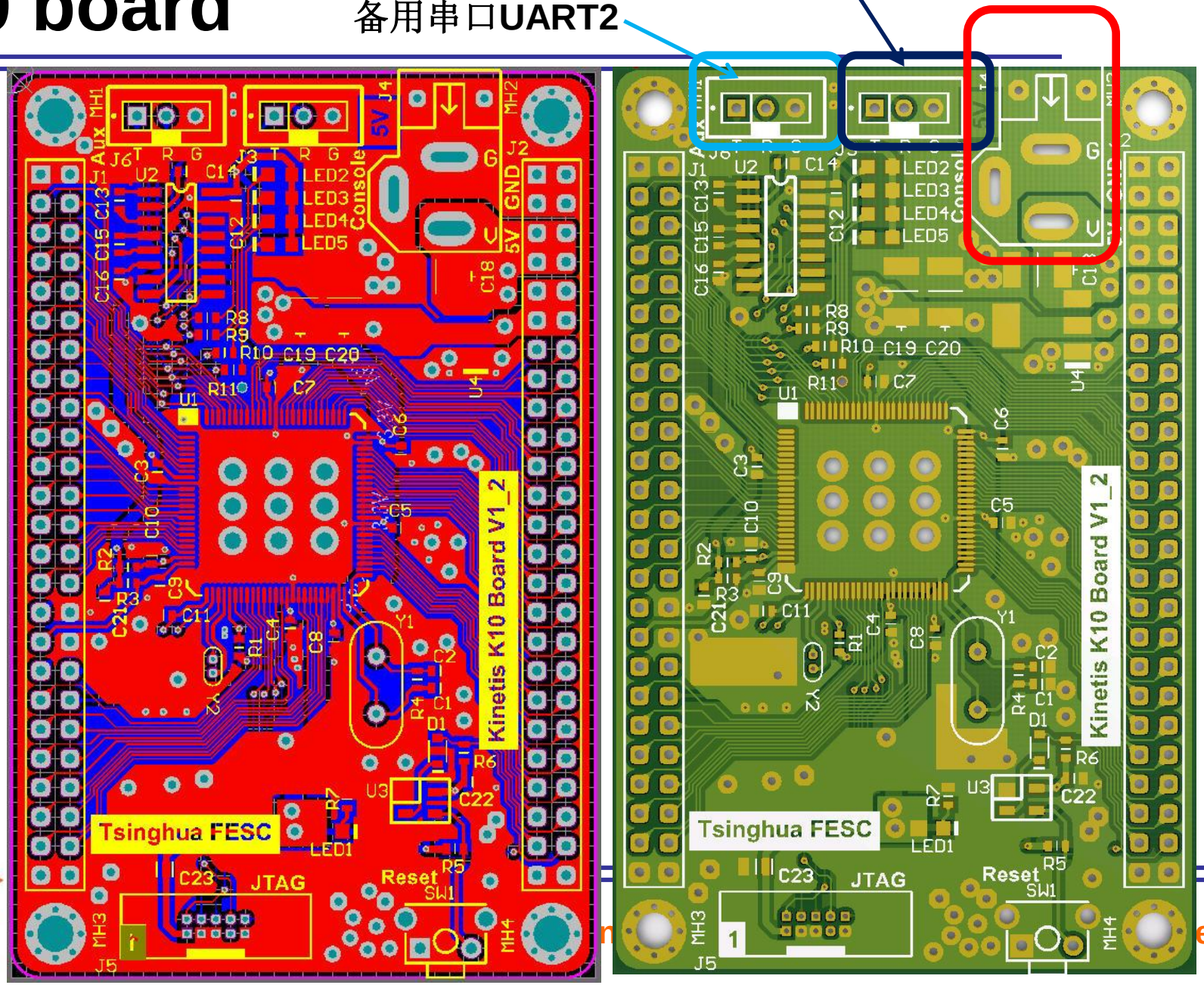
Revision:

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)

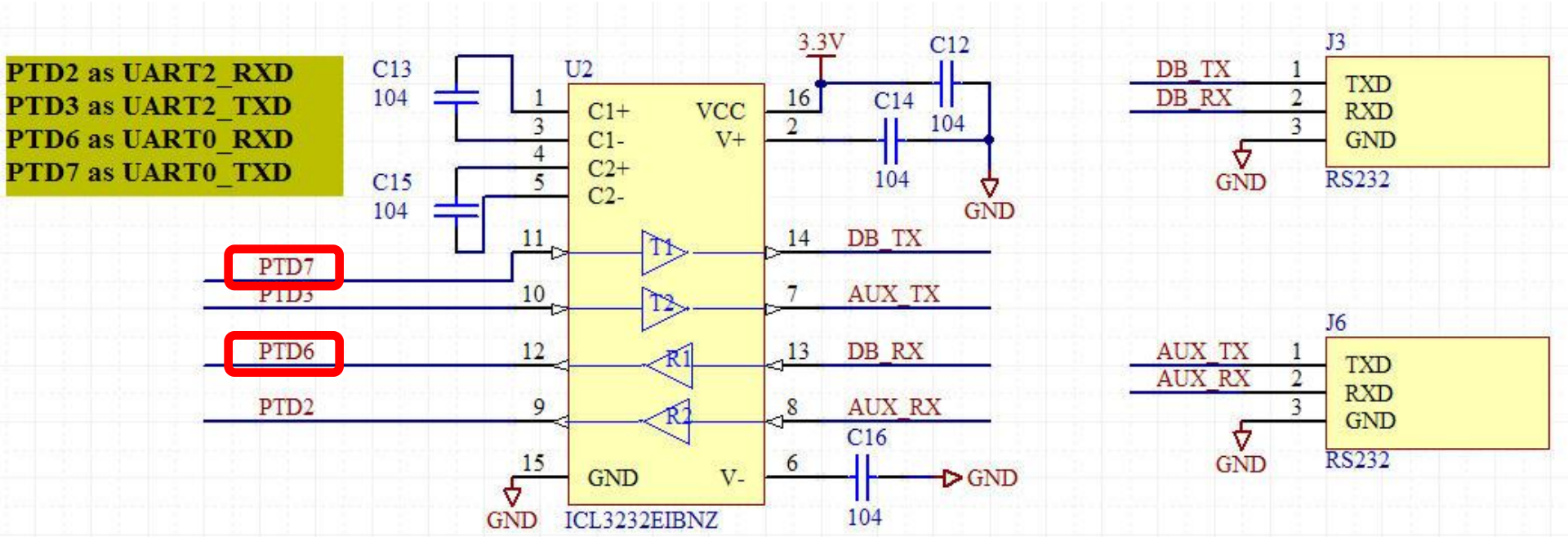
K10 board


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

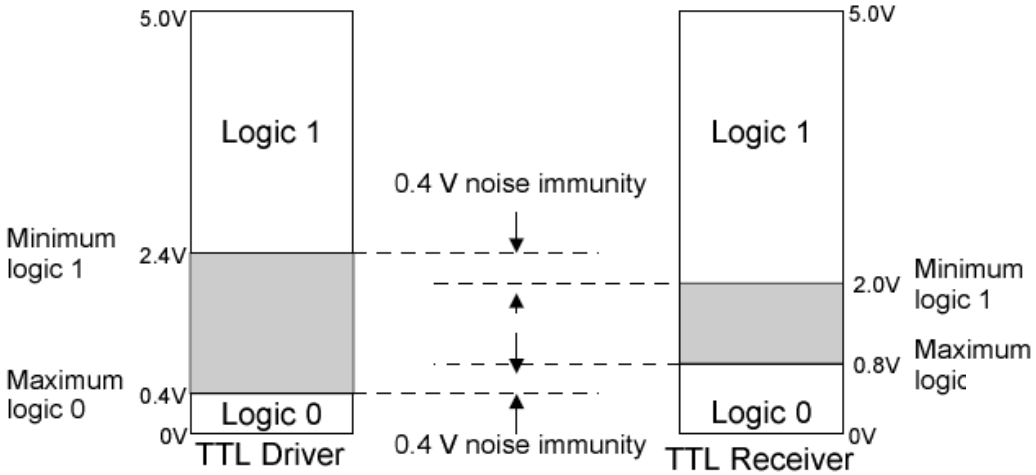
K10 board

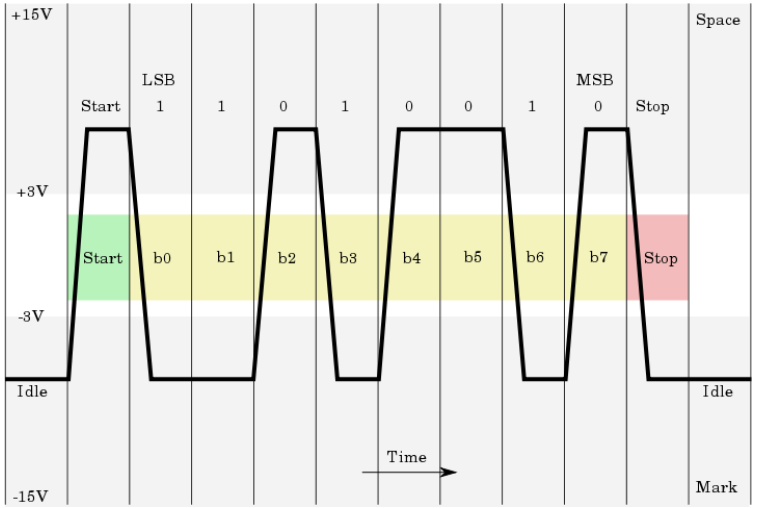

调试用串口UART0
电源 直流5V 100mA
备用串口UART2
Kinetis K10 Board V1_2
Tsinghua FESC
JTAG
Reset
LED1
LED2
LED3
LED4
LED5
Console
5V
GND

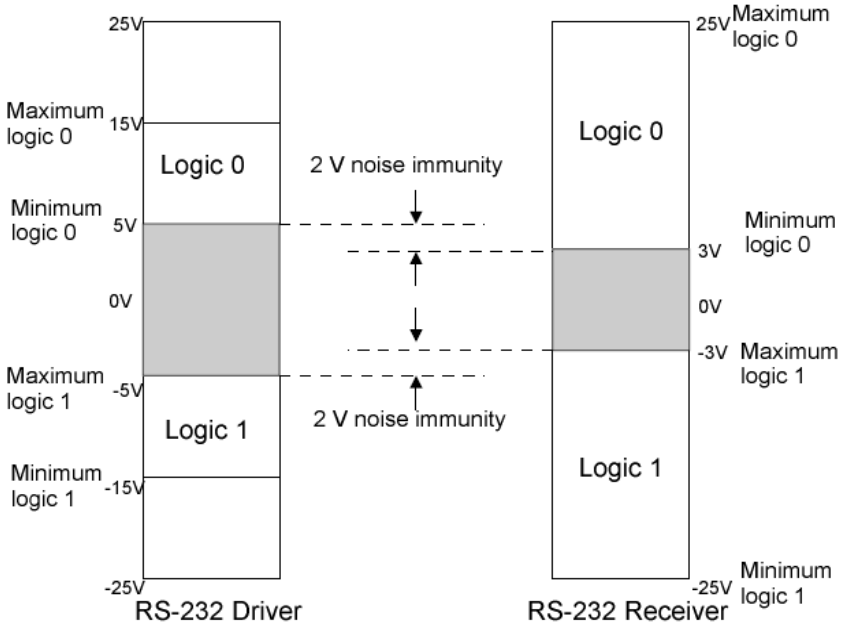
K10 board RS232 & JTAG



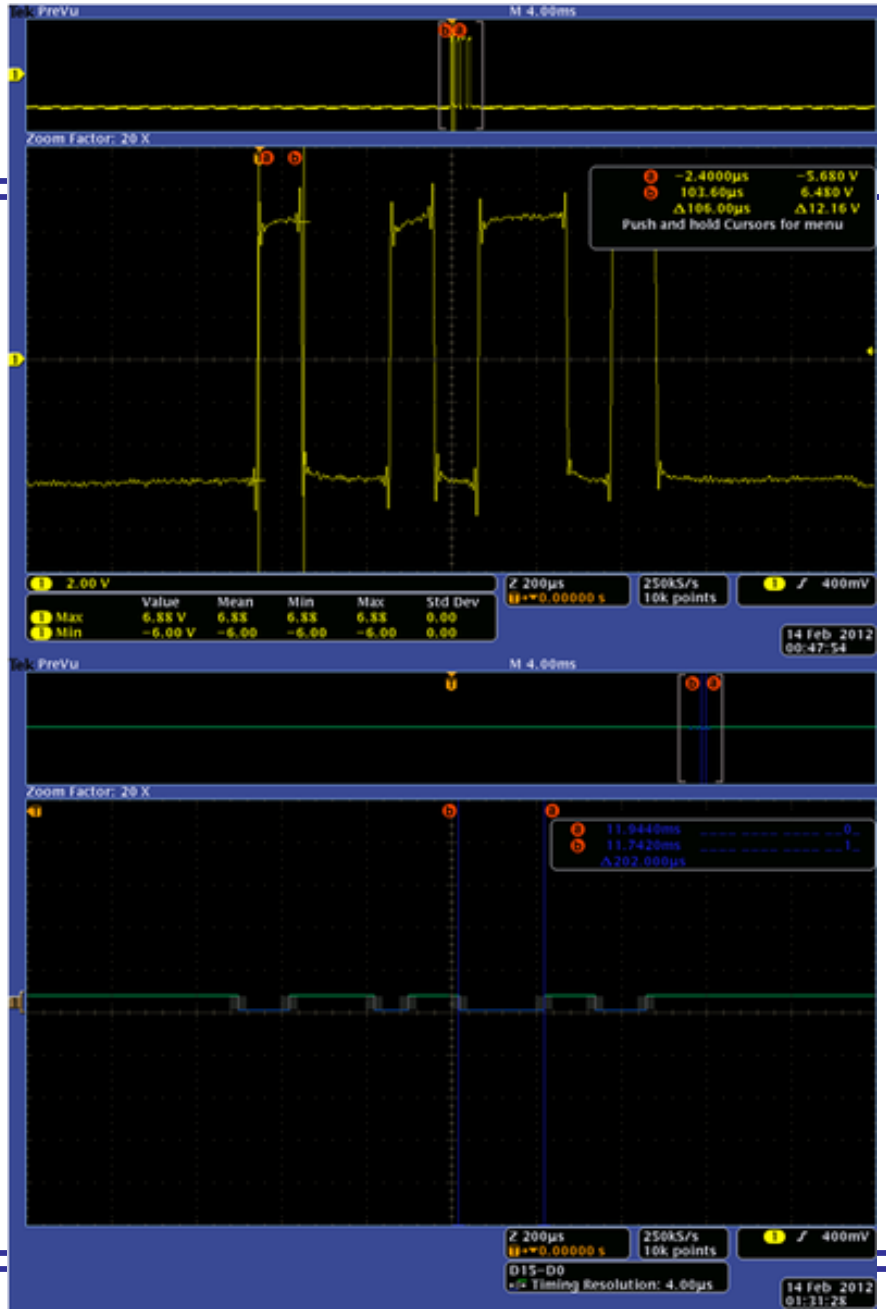
K10 TTL & RS232





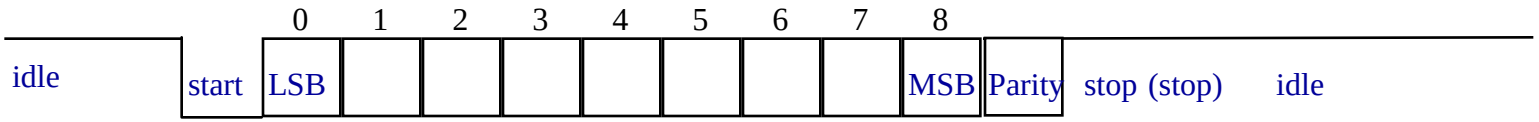


K10 TTL & RS232



K10 RS232 Frame

Frame format



- NRZ (not return zero) encoding
- Several configurable parameters
  - Baud rate, #data bits, parity, #stop bits
- Nomenclature: 9600,N81
- LSB- Least significant bit
- MSB- Most significant bit

K10 UART Signal

Signal	I/O	Description	
$\overline{CTS}$	I	Clear to send. Indicates whether the UART can start to transmit data when flow control is enabled.	
		State meaning	Asserted—Data transmission can start. Negated—Data transmission can not start.
		Timing	Assertion—When transmitting device's $\overline{RTS}$ asserts. Negation—When transmitting device's $\overline{RTS}$ deasserts.
$\overline{RTS}$	O	Request to send. When driven by the receiver, indicates whether the UART is ready to receive data. When driven by the transmitter, can enable an external transceiver during transmission.	
		State Meaning	Asserted—When driven by the receiver, ready to receive data. When driven by the transmitter, enable the external transmitter. Negated—When driven by the receiver, not ready to receive data. When driven by the transmitter, disable the external transmitter.
		Timing	Assertion—Can occur at any time; can assert asynchronously to the other input signals. Negation—Can occur at any time; can deassert asynchronously to the other input signals.
RXD	I	Receive data. Serial data input to receiver.	
		State meaning	Whether RXD is interpreted as a '1' or '0' depends on the bit encoding method along with other configuration settings.
		Timing	Sampled at a frequency determined by the module clock divided by the baud rate.
TXD	O	Transmit data. Serial data output from transmitter.	
		State meaning	Whether TXD is interpreted as a '1' or '0' depends on the bit encoding method along with other configuration settings.
		Timing	Driven at the beginning or within a bit time according to the bit encoding method along with other configuration settings. Otherwise, transmissions are independent of reception timing.

K10 SIM

SIM memory map

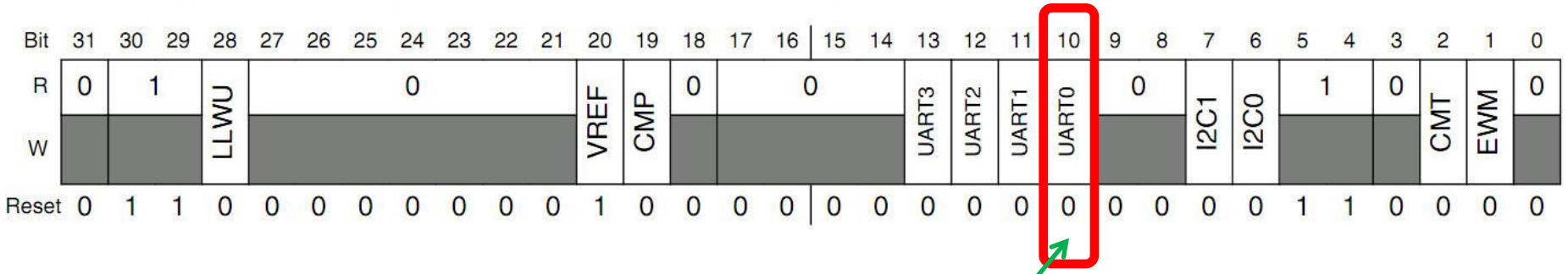
Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_7000	System Options Register 1 (SIM_SOPT1)	32	R/W	Undefined	12.2.1/250
4004_8004	System Options Register 2 (SIM_SOPT2)	32	R/W	0000_1000h	12.2.2/252
4004_800C	System Options Register 4 (SIM_SOPT4)	32	R/W	0000_0000h	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

Detail in Page 265 of K10P100M100SF2RM.pdf

Address: SIM_SCGC4 is 4004_7000h base + 1034h offset = 4004_8034h



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	1		LLWU	0							VREF	CMP	0	0				UART3	UART2	UART1	UART0	0		I2C1	I2C0	1		0	CMT	EWM	0
W																																
Reset	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

UART0 Clock Gate Control

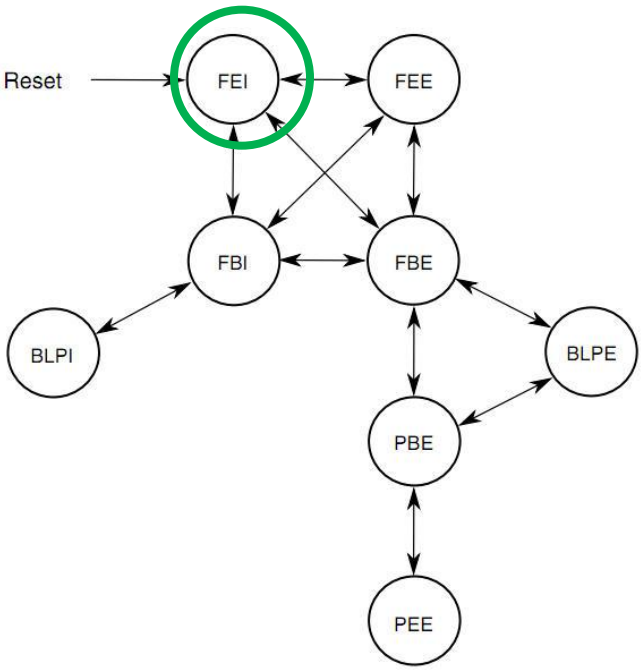
This bit controls the clock gate to the UART0 module.

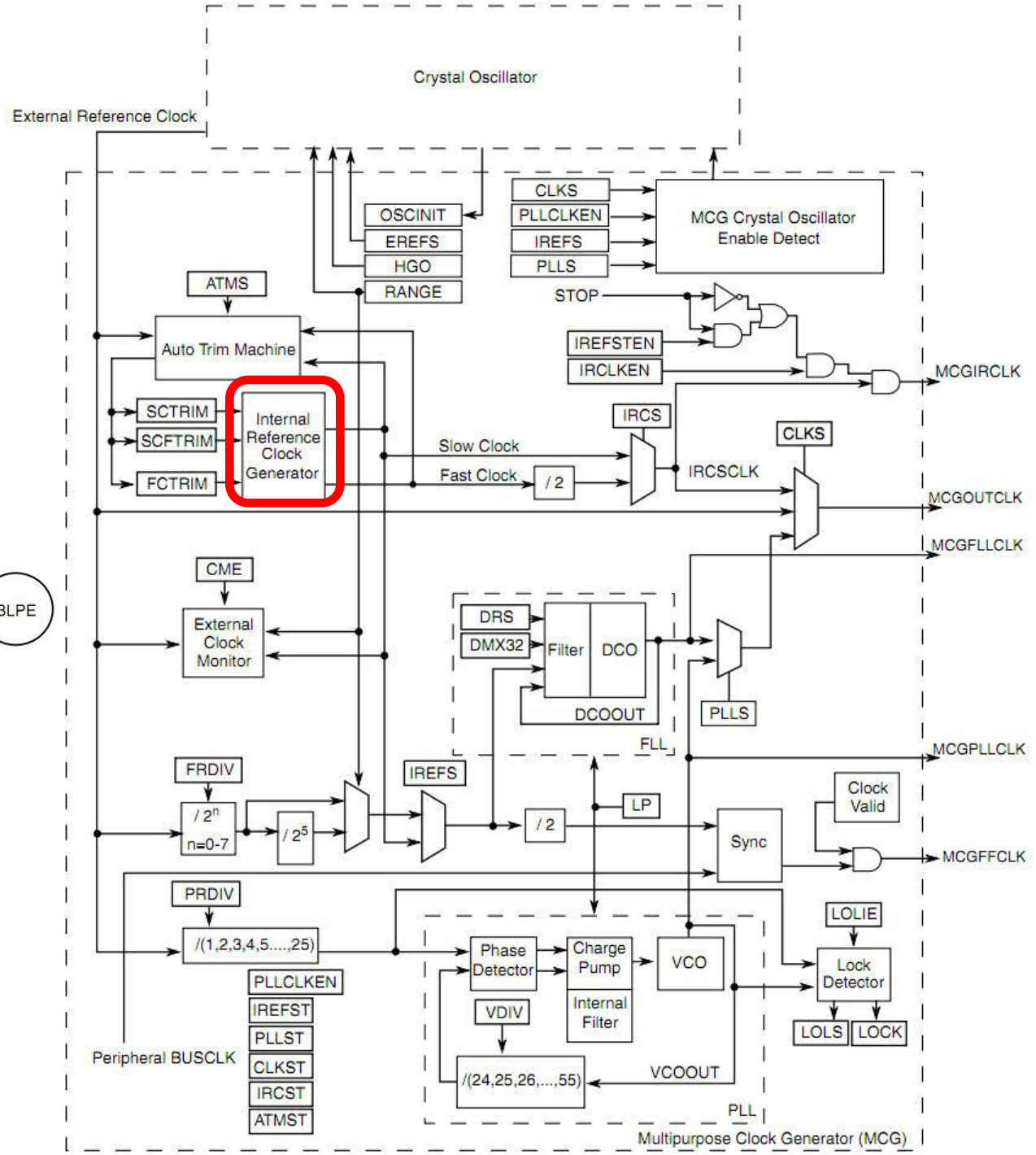
0 Clock disabled

1 Clock enabled

```
/* Enable the clock to the selected UART */
SIM_SCGC4 |= SIM_SCGC4_UART0_MASK;
```

K10 MCG


Reset
FEI
FEE
FBI
FBE
BLPI
BLPE
PBE
PEE


Crystal Oscillator
External Reference Clock
OSCINIT
EREFS
HGO
RANGE
CLKS
PLLCLKEN
IREFS
PLLS
MCG Crystal Oscillator Enable Detect
ATMS
Auto Trim Machine
STOP
IREFSTEN
IRCLKEN
MCGIRCLK
SCTRIM
SCFTRIM
FCTRIM
Internal Reference Clock Generator
IRCS
Slow Clock
Fast Clock
/ 2
IRCSCLK
CLKS
MCGOUTCLK
MCGFLLCLK
CME
External Clock Monitor
DRS
DMX32
Filter
DCO
DCOOUT
FLL
PLLS
MCGPLLCLK
FRDIV
/ 2^n n=0-7
/ 2^5
IREFS
/ 2
LP
Clock Valid
Sync
MCGFFCLK
PRDIV
/(1,2,3,4,5....,25)
Phase Detector
Charge Pump
VCO
LOLIE
Lock Detector
VDIV
Internal Filter
LOLS
LOCK
PLLCLKEN
IREFST
PLLST
CLKST
IRCST
ATMST
Peripheral BUSCLK
/(24,25,26,...,55)
VCOOUT
PLL
Multipurpose Clock Generator (MCG)

FESC

K10 MCG

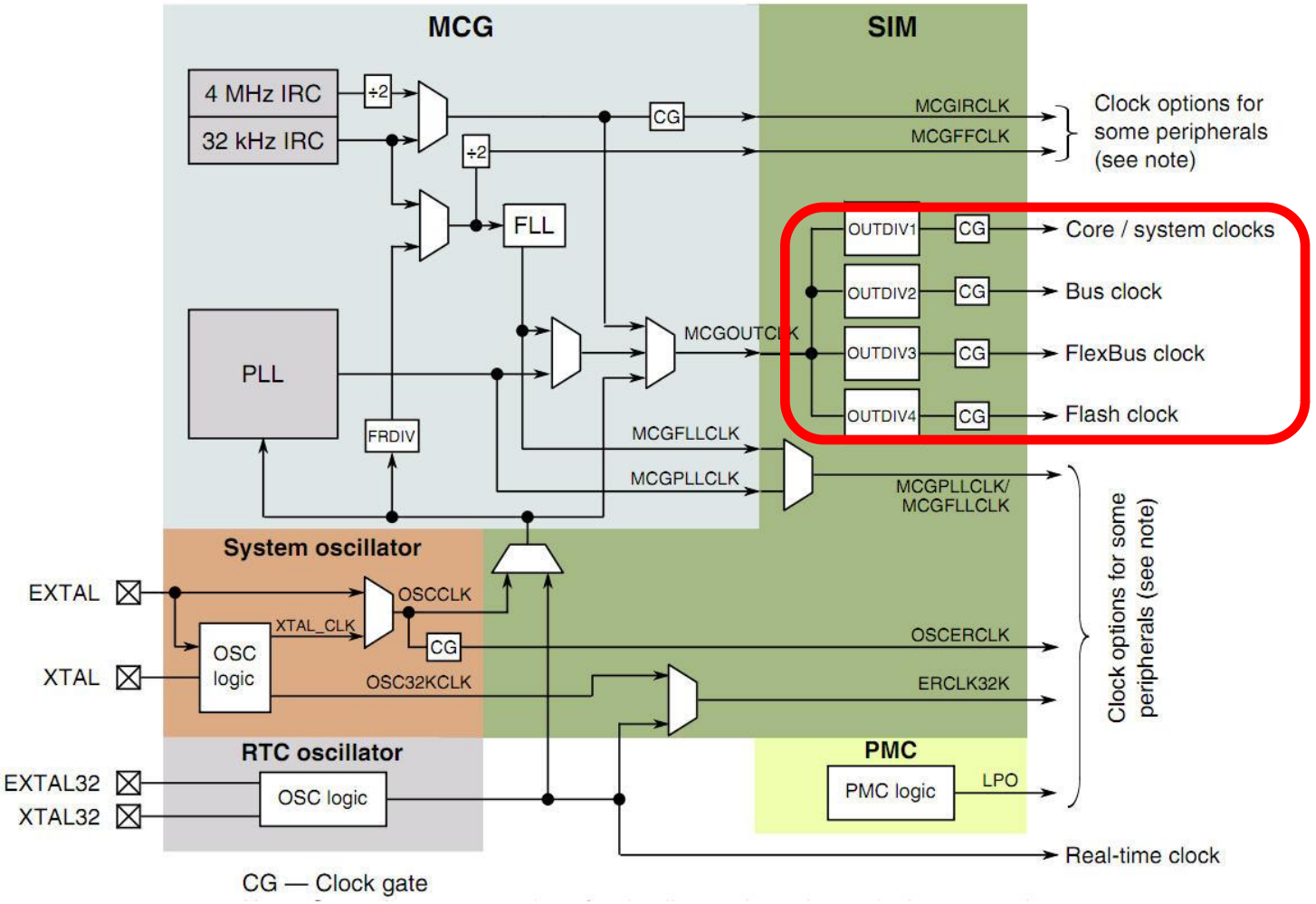
Detail in Page 538 of K10P100M100SF2RM.pdf

Use the default 20MHz as clock of UART module

Clock Mode	$f_{MCGOUTCLK}$[1]	Note
FEI (FLL engaged internal)	(f_{int} * F)	Typical $f_{MCGOUTCLK}$ = 20 MHz immediately after reset.
FEE (FLL engaged external)	(f_{ext} / FLL_R) *F	f_{ext} / FLL_R must be in the range of 31.25 kHz to 39.0625 kHz
FBE (FLL bypassed external)	f_{ext}	f_{ext} / FLL_R must be in the range of 31.25 kHz to 39.0625 kHz
FBI (FLL bypassed internal)	f_{int}	Typical f_{int} = 32 kHz
PEE (PLL engaged external)	(f_{ext} / PLL_R) * M	f_{ext} / PLL_R must be in the range of 2 – 4 MHz
PBE (PLL bypassed external)	f_{ext}	f_{ext} / PLL_R must be in the range of 2 – 4 MHz
BLPI (Bypassed low power internal)	f_{int}	
BLPE (Bypassed low power external)	f_{ext}	

K10 MCG

Detail in Page 160 of K10P100M100SF2RM.pdf



K10 IO 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 IO 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

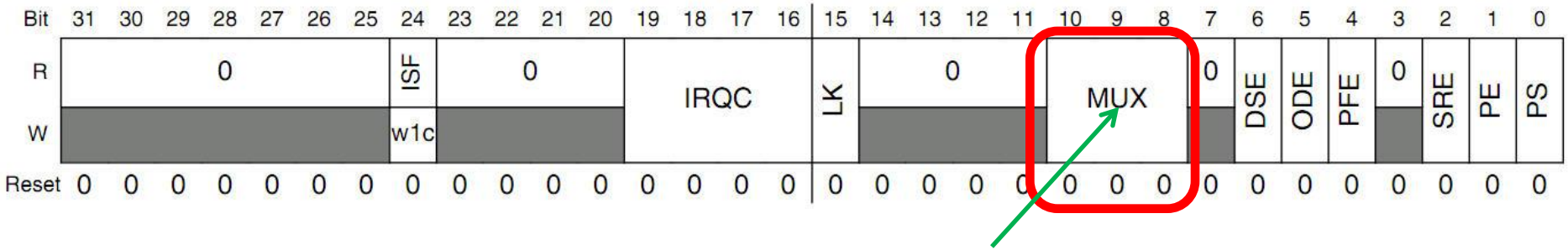
Pull Enable

Pull Select
Pull Up
Pull Down

K10 IO 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). UART ?
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 UART 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

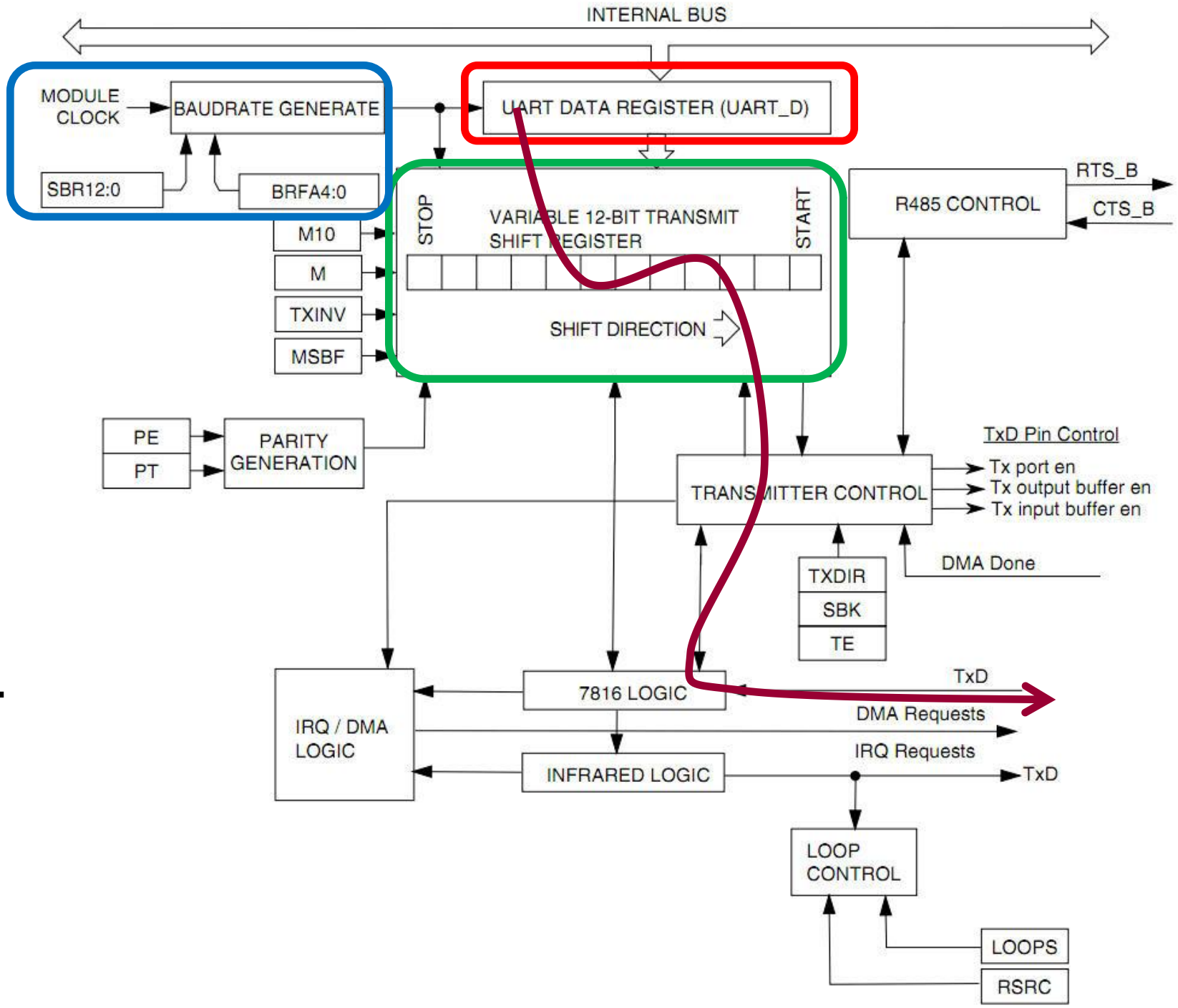
100 LQFP	Pin Name	Default	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5	ALT6	ALT7	EzPort
								FB_BE15_8_BLS23_16_b			
93	PTD0			PTD0	SPI0_PCS0	UART2_RTS_b		FB_ALE/FB_CS1_b/FB_TS_b			
94	PTD1	/ADC0_SE5b	/ADC0_SE5b	PTD1	SPI0_SCK	UART2_CTS_b		FB_CS0_b			
95	PTD2			PTD2	SPI0_SOUT	UART2_RX		FB_AD4			
96	PTD3			PTD3	SPI0_SIN	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 UART PCR

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

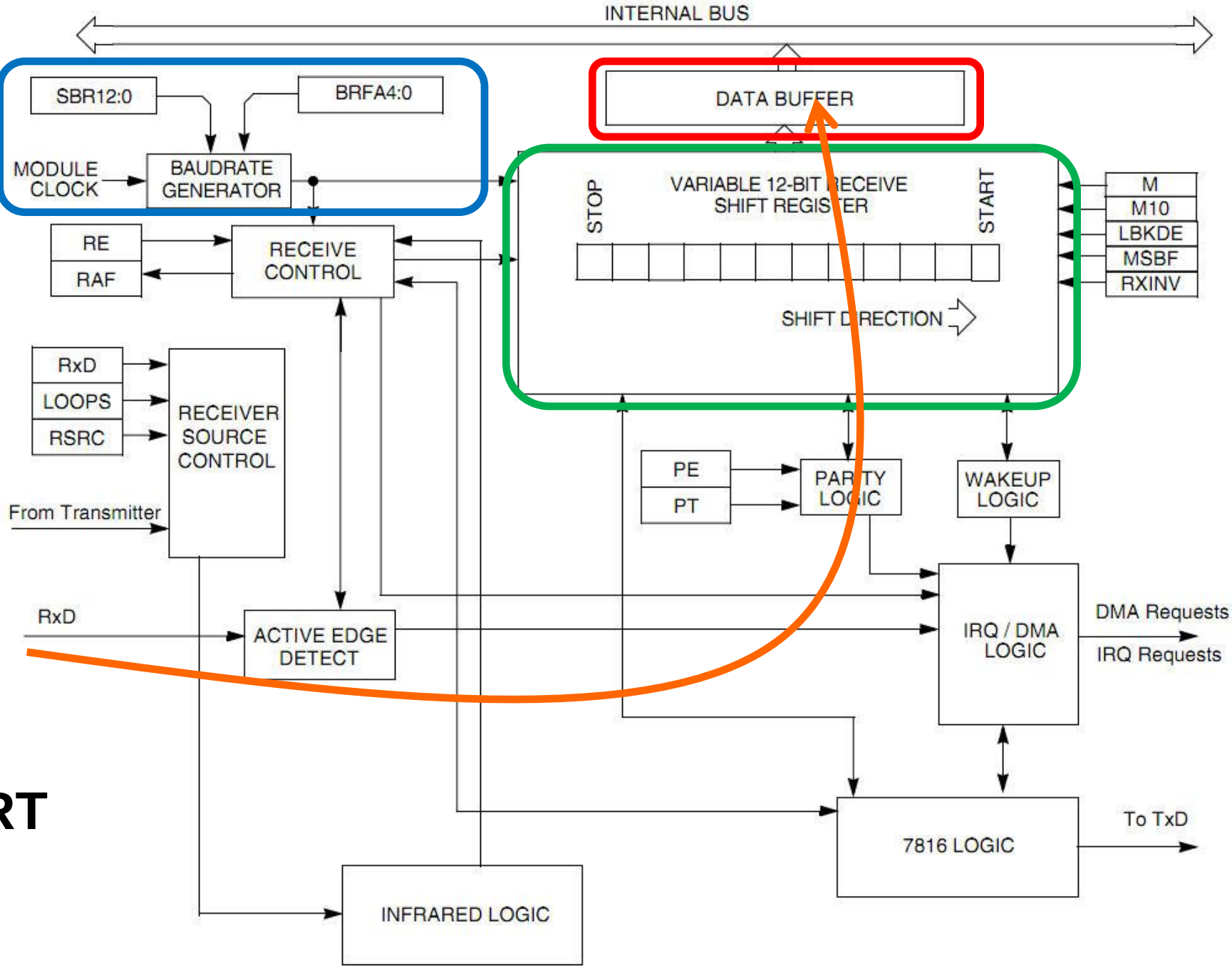
```
/* Enable the UART0_TXD function on PTD6 */
 PORTD_PCR6 = PORT_PCR_MUX(0x3); // UART is alt3 function for this pin

 /* Enable the UART0_RXD function on PTD7 */
 PORTD_PCR7 = PORT_PCR_MUX(0x3); // UART is alt3 function for this pin
```

K10 UART



FESC

K10
UART



FESC

K10 UART

Detail in Page 1205 of K10P100M100SF2RM.pdf

Control Mode:

UARTx_BDH	UART Baud Rate Register High
UARTx_BDL	UART Baud Rate Register Low
UARTx_C1	UART Control Register 1
UARTx_C2	UART Control Register 2
UARTx_C3	UART Control Register 3
UARTx_C4	UART Control Register 4
UARTx_C5	UART Control Register 5
UARTx_S1	UART Status Register 1
UARTx_S2	UART Status Register 2

K10 UART BDH:L

Detail in Page 1217 of K10P100M100SF2RM.pdf

BDH

Bit	7	6	5	4	3	2	1	0
Read	LBKDIE	RXEDGIE	0	SBR				
Write								
Reset	0	0	0	0	0	0	0	0

BDL

Bit	7	6	5	4	3	2	1	0
Read	SBR							
Write								
Reset	0	0	0	0	0	1	0	0

C4

Bit	7	6	5	4	3	2	1	0
Read	MAEN1	MAEN2	M10	BRFA				
Write								
Reset	0	0	0	0	0	0	0	0

UART Baud rate = UART module clock/(16 x (SBR[12:0] + BRFD))

BRFD = 32 x BRFA[5:0]

K10 UART Baud rate calculation

```
/* Calculate baud settings */
unsigned short uartclk_khz = 20000;
unsigned short baud = 9600;
unsigned short sbr, temp, brfa;

sbr = (unsigned short)((uartclk_khz *1000)/(baud * 16));

UART0_BDH = (unsigned char)((sbr & 0x1F00) >> 8);
UART0_BDL = (unsigned char)(sbr & 0x00FF);

/* Determine if a fractional divider is needed to get closer to the baud rate */
brfa = (((uartclk_khz*32000)/(baud * 16)) - (sbr * 32));

UART0_C4 = (unsigned char)(brfa & 0x001F);
```

Detail in Page 1273 of K10P100M100SF2RM.pdf

K10 UART C1

Detail in Page 1220 of K10P100M100SF2RM.pdf

Bit	7	6	5	4	3	2	1	0
Read Write	LOOPS	UARTSWAI	RSRC	M	WAKE	ILT	PE	PT
Reset	0	0	0	0	0	0	0	0

LOOP mode select

9-bit or 8-bit Mode Select

Parity Function

Parity Type: Even or Odd

```
/* Configure the UART for 8-bit mode, no parity */
   UART0_C1 = 0;
```

K10 UART C2

Detail in Page 1221 of K10P100M100SF2RM.pdf

Bit	7	6	5	4	3	2	1	0
Read Write	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
Reset	0	0	0	0	0	0	0	0

TXD Interrupt Enable

TXD Complete Interrupt Enable

RXD Interrupt Enable

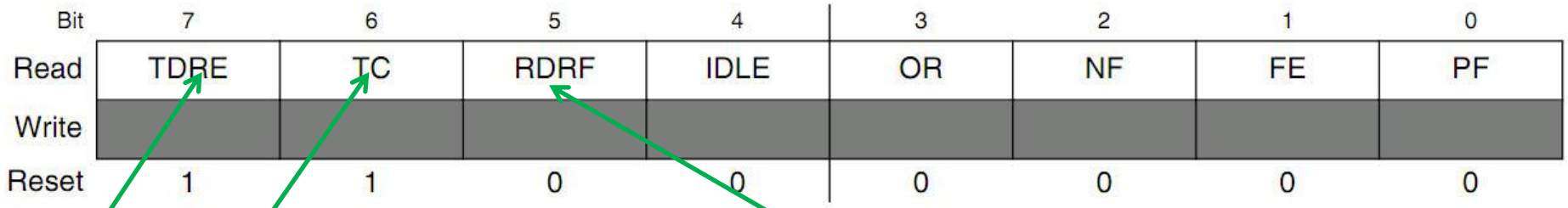
TXD Enable

RXD Enable

```
/* Enable receiver and transmitter */
UART0_C2 |= (UART_C2_TE_MASK | UART_C2_RE_MASK );
```

K10 UART S1

Detail in Page 1223 of K10P100M100SF2RM.pdf



Bit	7	6	5	4	3	2	1	0
Read	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF
Write								
Reset	1	1	0	0	0	0	0	0

TXD Data Buffer Empty Flag

TXD Complete Flag

RXD Data Buffer Full Flag

```
void uart0_putchar (unsigned char data)
{
   /* Wait until space is available in the FIFO */
   while(!(UART0_S1 & UART_S1_TDRE_MASK));

   /* Send the character */
   UART0_D = data;
 }
```

K10 UART S1

Detail in Page 1223 of K10P100M100SF2RM.pdf

Bit	7	6	5	4	3	2	1	0
Read	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF
Write								
Reset	1	1	0	0	0	0	0	0

TXD Data Buffer Empty Flag

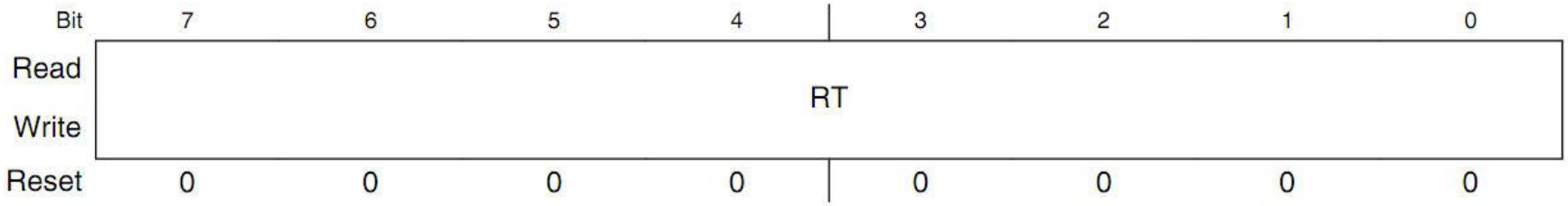
TXD Complete Flag

RXD Data Buffer Full Flag

```
unsigned char uart0_getchar (void)
{
   /* Wait until character has been received */
   while (!(UART0_S1 & UART_S1_RDRF_MASK));

   /* Return the 8-bit data from the receiver */
   return UART0_D;
}
```

K10 UART Data

Detail in Page 1231 of K10P100M100SF2RM.pdf



UARTx_D field descriptions

Field	Description
7–0 RT	Reads return the contents of the read-only receive data register and writes go to the write-only transmit data register.

K10 UART Demo Code

```
unsigned char uart0_getchar (void)
{
/* Wait until character has been received */
while (!(UART0_S1 & UART_S1_RDRF_MASK));
/* Return the 8-bit data from the receiver */
return UART0_D;
}

void uart0_putchar (unsigned char data)
{
/* Wait until space is available in the FIFO */
while(!(UART0_S1 & UART_S1_TDRE_MASK));
/* Send the character */
UART0_D = data;
}
```

K10 UART Demo Code

```
int main(void)
{
        int counter = 0;    /* Calculate baud settings */
        unsigned short uartclk_khz = 20000;
        unsigned short baud = 9600;
        unsigned short sbr, temp, brfa;
        unsigned char tom;
        SIM_SCGC4 |= SIM_SCGC4_UART0_MASK;          /* Enable the clock to the selected UART */
        SIM_SCGC5 |= SIM_SCGC5_PORTD_MASK;          /* Turn on all port clocks */
        /* Enable the UART0_TXD function on PTD6 */
        PORTD_PCR6 = PORT_PCR_MUX(0x3); // UART is alt3 function for this pin
        /* Enable the UART0_RXD function on PTD7 */
        PORTD_PCR7 = PORT_PCR_MUX(0x3); // UART is alt3 function for this pin
        sbr = (unsigned short)((uartclk_khz *1000)/(baud * 16));
        UART0_BDH = (unsigned char)((sbr & 0x1F00) >> 8);
        UART0_BDL = (unsigned char)(sbr & 0x00FF);
        /* Determine if a fractional divider is needed to get closer to the baud rate */
        brfa = (((uartclk_khz*32000)/(baud * 16)) - (sbr * 32));
        UART0_C4 = (unsigned char)(brfa & 0x001F);
        UART0_C1 = 0;               /* Configure the UART for 8-bit mode, no parity */
        UART0_C2 |= (UART_C2_TE_MASK | UART_C2_RE_MASK );      /* Enable receiver and transmitter */
        for(;;)
        {
                counter++;
                tom = uart0_getchar();
                uart0_putchar(tom + 1);
        }
        return 0;
}
```