Preliminary

CANTUS

- *UART* -

32bits EISC Microprocessor *CANTUS*

Ver 1.0
October 8, 2009

Advanced Digital Chips Inc.

History

2009-10-08	Created Preliminary Specification

CANTUS Evaluation Board Application Note : #0002 UART

Office

8th Floor, KookMin 1 Bldg.,

1009-5, Daechi-Dong, Gangnam-Gu, Seoul, 135-280, Korea

Tel : + 82-2-2107-5800

Fax : + 82-2-571-4890

URL : http://www.adc.co.kr

— Table of Contents —

— List of Figures —

— List of Tables —

1 Summary

이 문서는 CANTUS SDK의 UART에 대한 Application Note이다.

UART Project는 CANTUS의 UART Controller를 이용하여 PC와 직렬 비동기 통신을 하게 된다. 통신이 연결되면 사용자가 하이퍼터미널에서 입력한 키를 다시 PC로 전송하여 화면에 출력하는 예제이다.

또한 모든 Application은 Debug용도로 UART 7번 Channel을 사용하고 있다.

2 Register Set

2.1 Register Set Flow Chart

CANTUS의 UART를 사용하기 위해선 다음과 같은 순서로 Register를 설정한다.

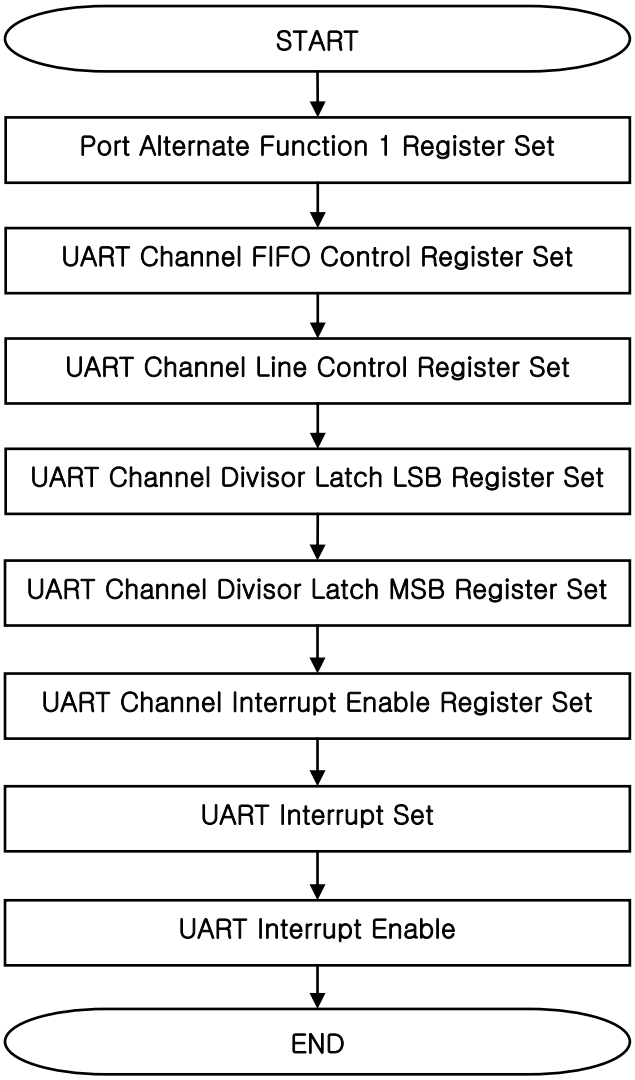

그림 2-1 Register Set Flow Chart

2.2 Port Alternate Function Register

UART Channel 7은 8, 9번 PIN을 사용한다. 이들 PIN은 아래 표 2-1과 같이 다른 Function을 공유 하고 있어 UART Channel 7을 사용하기 위해서는 Port Alternate Function 1 Register에서 8번 9번 PIN을 TX[7], RX[7]로 설정하여야 한다.[1]

표 2-1 Port 1 Alternate Function

Group	Index	Pin	1st	2nd	3rd	4th(default)
PAF1	**0**	2	TX[4]	KEY_O[0]	AD[8]	P1.0
	1	3	RX[4]	KEY_I[0]	AD[9]	P1.1
	2	4	TX[5]	KEY_O[1]	AD[10]	P1.2
	3	5	RX[5]	KEY_I[1]	AD[11]	P1.3
	4	6	TX[6]	KEY_O[2]	AD[12]	P1.4
	5	7	RX[6]	KEY_I[2]	AD[13]	P1.5
	6	8	TX[7]	KEY_O[3]	AD[14]	P1.6
	7	9	RX[7]	KEY_I[3]	AD[15]	P1.7

표 2-2 Port Alternate Function 1 Register (PAF1)

Address : 0x8002_0024

Bit	R/W	Description	Default Value
31 : 16	R	Reserved	-
15 : 14	R/W	P1.7 : P1.7 Port Selection bit 00 : RX[7] 01 : KEYI[3] 10 : AD[15] 11 : P1.7	11
13 : 12	R/W	P1.6 : P1.6 Port Selection bit 00 : TX[7] 01 : KEYO[3] 10 : AD[14] 11 : P1.6	11
11 : 10	R/W	P1.5 : P1.5 Port Selection bit 00 : RX[6] 01 : KEYI[2] 10 : AD[13] 11 : P1.5	11
9 : 8	R/W	P1.4 : P1.4 Port Selection bit 00 : TX[6] 01 : KEYO[2] 10 : AD[12] 11 : P1.4	11
7 : 6	R/W	P1.3 : P1.3 Port Selection bit 00 : RX[5] 01 : KEYI[1] 10 : AD[11] 11 : P1.3	11
5 : 4	R/W	P1.2 : P1.2 Port Selection bit 00 : TX[5] 01 : KEYO[1] 10 : AD[10] 11 : P1.2	11
3 : 2	R/W	P1.1 : P1.1 Port Selection bit 00 : RX[4] 01 : KEYI[0] 10 : AD[9] 11 : P1.1	11
1 : 0	R/W	P1.0 : P1.0 Port Selection bit 00 : TX[4] 01 : KEYO[0] 10 : AD[8] 11 : P1.0	11

[1] CANTUS Datasheet의 8 GPIO 참조.

2.3 UART Channel FIFO Control Register

표 2-3 UART Channel FIFO Control Register 7 (U7FC)

Address : 0x8002_1868

Bit	*R/W*	*Description*	*Default Value*
31 : 8	R	Reserved.	-
7 : 6	RW	RFTL : Receiver FIFO Trigger Level 00 : 1 Byte 01 : 4 Byte 10 : 8 Byte 11 : 16 Byte	00
5 : 3	R	Reserved	-
2	RW	XFR : XMIT FIFO Reset XFR가 "1" 일 때, XMIT FIFO 내의 모든 데이터는 Reset 된다. 그러나 Shift Register 내의 데이터는 Reset 되지 않는다.	0
1	RW	RFR : RCVR FIFO Reset RFR가 "1" 일 때, RCVR FIFO 내의 모든 데이터는 Reset 된다, 그러나 Shift Register 내의 데이터는 Reset 되지 않는다.	0
0	RW	FIFOEN : FIFO Enable Bit 0 : 16450 UART Mode 1 : Enables FIFO	0

*** DLAB가 "0" 일 때는 Write Mode 이고, DLAB가 "1" 일 때는 Read Mode 이다.

2.4 UART Channel Line Control Register

표 2-4 UART Channel Line Control Register 7 (U7LC)

Address : 0x8002_186C

Bit	*R/W*	*Description*	*Default Value*
31 : 8	R	Reserved.	-
7	RW	DLAB : Divisor Latch Access Bit DLAB이 "1" 일 때, Divisor Latch Registers의 Read/Write와 FIFO Control Register의 Read가 가능하다.	0
6	RW	SB : Set Break SB가 "1" 일 때, Serial Data Output에 Logic "0"이 출력된다. SB는 내부 Transmitter Logic에는 영향을 미치지 않으며, 단지 Serial Output에만 영향을 미친다.	0
5	RW	SP : Stick Parity 0 : Disables Stick Parity 1 : PEN, EPS, SP가 "1"일 때, Parity Bit "0" PEN, SP가 "1"이고, EPS가 "0" 일 때, Parity Bit "1"	0
4	RW	EPS : Even Parity Select 0 : Select Odd Parity 1 : Select Even Parity	0
3	RW	PEN : Parity Enable Bit 0 : Disables Parity 1 : Enables Parity	0
2	RW	STB : Number of Stop Bit 0 : 1 Stop bit 1 : 2 Stop bits(만약, WLS Bit에서 5 Bits/Character를 선택했다면, 1.5 Stop bits 을 갖는다.)	0
1 : 0	RW	WLS : Word Length Select 00 : 5 Bits/Character 01 : 6 Bits/Character 10 : 7 Bits/Character 11 : 8 Bits/Character	00

2.5 UART Channel Divisor Latch LSB Register

표 2-5 UART Channel Divisor Latch LSB Register 7 (U7DLL)

Address : 0x8002_1860

Bit	R/W	Description	Default Value
31: 8	R	Reserved.	-
7 : 0	RW	Divisor Latch Least Significant Byte	0x00

*** DLAB가 "1" 일 때 Access 가능하다.

2.6 UART Channel Divisor Latch MSB Register

표 2-6 UART Channel Divisor Latch MSB Register 7 (U7DLM)

Address : 0x8002_1864

Bit	R/W	Description	Default Value
31: 8	R	Reserved.	-
7 : 0	RW	Divisor Latch Most Significant Byte	0x00

*** DLAB가 "1" 일 때 Access 가능하다.

Baud Rate은 아래 식으로 계산된다.

$$UART\ \ Baud\ \ Rate = \frac{f_{PCLK}}{16\ \times\ UDL}$$

UART Divisor Latch Value (UDL) = UDLM[7:0] << 8 + UDLL[7:0]

표 2-7 UART Baud Rate

f_{PCLK}(MHz)		1.024	2.048	5.6448	11.2896	24.0	48.0
2400 bps	UDL	27	53	147	294	625	1250
	ERR(%)	1.23	0.63	0.00	0.00	0.00	0.00
4800 bps	UDL	-	27	74	147	313	625
	ERR(%)	-	1.23	0.68	0.00	0.16	0.00
9600 bps	UDL	-	-	37	74	156	313
	ERR(%)	-	-	0.68	0.68	0.16	0.16
14400 bps	UDL	-	9	25	49	104	208
	ERR(%)	-	1.23	2.00	0.00	0.16	0.16
19200 bps	UDL	-	-	18	37	78	156
	ERR(%)	-	-	2.08	0.68	0.16	0.16
38400 bps	UDL	-	-	9	18	39	78
	ERR(%)	-	-	2.08	2.08	0.16	0.16
57600 bps	UDL	-	-	6	12	26	52
	ERR(%)	-	-	2.08	2.08	0.16	0.16
115200bps	UDL	-	-	3	6	13	26
	ERR(%)	-	-	2.08	2.08	0.16	0.16

*** ERR 이 2.2% 이상에서는 UART 동작의 안정성을 보장 받을 수 없다.

2.7 UART Channel Interrupt Enable Register

표 2-8 UART Channel Interrupt Enable Register 7 (U7IE)

Address : 0x8002_1864

Bit	R/W	Description	Default Value
31: 3	R	Reserved.	-
2	RW	RLSIEN : Receiver Line Status Interrupt Enable bit 0 : Disable 1 : Enable	0
1	RW	THEIEN : Transmitter Holding Empty Interrupt Enable bit 0 : Disable 1 : Enable	0
0	RW	RDAIEN : Received Data Available Interrupt Enable bit 0 : Disable 1 : Enable	0

*** DLAB가 "0" 일 때 Access 가능하다.

2.8 UART Interrupt Set

UartConfig() **함수에서 아래 함수를 호출하여 수행한다.**

BOOL setinterrupt(INTERRUPT_TYPE intnum,void (*fp)());

\Cantuslib\interrupt.c

2.9 UART Interrupt Enable

UartConfig() **함수에서 아래 함수를 호출하여 수행한다.**

void EnableInterrupt(INTERRUPT_TYPE num,BOOL b);

\Cantuslib\interrupt.c

2.10 UART Channel Receiver Buffer Register

UART로 수신한 Data를 저장하는 Buffer

표 2-9 UART Channel Receiver Buffer Register 7 (U7RB)

Address : 0x8002_1860

Bit	R/W	Description	Default Value
31: 8	R	Reserved.	-
7 : 0	R	Receive Buffer Data	-

*** DLAB가 "0" 일 때 Access 가능하다.

2.11 UART Channel Transmitter Holding Register

UART로 송신할 Data Buffer

표 2-10 UART Channel Transmitter Holding Register 7 (U7TH)

Address : 0x8002_1860

Bit	R/W	Description	Default Value
31: 8	W	Reserved.	-
7 : 0	W	Transmit Holding Data	-

*** DLAB가 "0" 일 때 Access 가능하다.

3 Function Set

3.1 Function Set Flow Chart

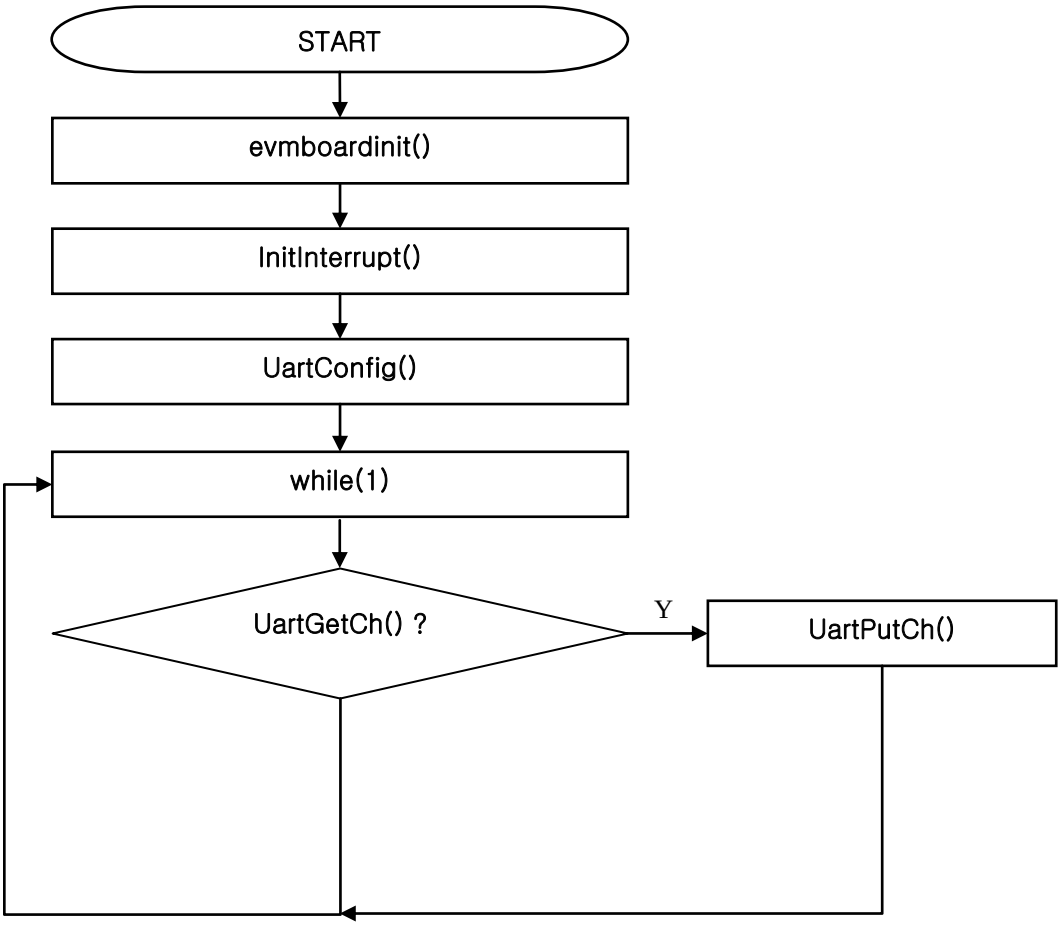

그림 3-1 Function Set Flow Chart

3.1.1 evmboardinit()

evmboardinit() 함수는 CANTUS Evaluation Board를 사용하기 위한 기본 Setting을 수행한다.

```
void evmboardinit()
{
	*(volatile U16*)0x80000404 = 0x0200;//BANK 1 8Bit
	*R_PAF0 = 0xAAAA; //SRAM interface address, data
	*R_PAF1 = 0x0d50; //uart4,KEYio1,KEYo2,,PIO1.5,UART7
	*R_PAF2 = 0xAAAA; /* nCS, nWE, nRE, ALE */
	*R_PAF3 = 0xAAA0; //EIRQ0-1,NDFL_*
	*R_PAF4 = 0xAAAA; // I2S, SDCD, TWI
	*R_PAF5 = 0xAAAA; //SDCD,I2S,Address
	*R_PAF6 = 0x0300; //PIO6.4 Amp Shutdown
	*R_P6oDIR |= (1<<4); // PIO6.4 output mode

	*(volatile unsigned char *)(0x60020000) = LCD_BACK|SDC_PWR_BIT; //LCD,SDC
	int j;
	for (j=0; j<1000; j++);
	*(volatile unsigned char *)(0x60020000) = LCD_BACK|SDC_PWR_BIT|LCD_PWR; //LCD,SDC
}
```

3.1.2 InitInterrupt()

InitInterrupt() 함수는 CANTUS의 Interrupt를 사용하기 위한 초기화 과정을 수행한다. 사용자는 별도의 설정을 할 필요가 없다.

3.1.3 UartConfig()

BOOL UartConfig(int ch, int baud, UART_DATABITS databits, UART_STOPBITS stopbit,
UART_PARITY parity);

UartConfig() 함수는 CANTUS의 UART를 사용하기 위한 Configuration 과정을 진행하는 함수로 2.2 ~ 2.9에 해당하는 Register Set을 모두 진행하기 때문에 사용자가 별도의 Register 설정 없이 Uart를 사용할 수 있다.

UART_DATABITS, UART_STOPBITS, UART_PARITY는 uart.h에 정의된 구조체이다.

- int ch : UART의 Channel 번호
- int baud : UART Channel ch의 baudrate를 설정한다. Ex) 115200 bps일 경우 115200
- UART_DATABITS databits : Data로 사용할 bit의 개수를 설정
- UART_STOPBITS stopbit : Stop bit의 개수를 설정
- UART_PARITY parity : Parity check를 설정(None, Odd, Even)

3.1.4 UartGetCh()

BOOL UartGetCh(int n, char* ch);

- Return : TRUE or FALSE
- int n : Data를 가져올 UART의 Channel 번호
- char *ch : 수신 Data를 저장할 변수의 주소

3.1.5 UartPutCh()

BOOL UartPutCh(int n, char ch);

- Return : TRUE or FALSE
- int n : 송신할 UART의 Channel 번호
- char ch : 송신할 Data

3.2 Uart.c

\ Cantuslib \ Uart.c

CANTUS의 UART에 관련된 함수의 원형과 Ring_buffer가 구현되어있다.
SDK에서 제공하는 Application은 Evaluation Board에 맞도록 Debug용으로 UART 7번이 설정되어 있다.

\ Cantuslib \ Cantus_lib_config.h

Cantuslib를 Build하기 위한 환경설정으로 UART와 관련된 부분을 다음과 같이 구성해 놓았다.

```
/*********************************************
Uart Config
*********************************************/
//#define UART0_ENABLE
//#define UART1_ENABLE
//#define UART2_ENABLE
//#define UART3_ENABLE
//#define UART4_ENABLE
//#define UART5_ENABLE
//#define UART6_ENABLE
#define UART7_ENABLE

#define UART_BUF_SIZE 512
#define CONFIG_UART_RX_INTERRUPT
//#define CONFIG_UART_TX_INTERRUPT
#define DEBUG_CHANNEL 7
#define CONFIG_DEBUGPRINTF_ENABLE //debugprintf funcion , if enabled, 1kbyte need
```

- DEBUG_CHANNEL
  debugstring() 함수와 debugprintf() 함수의 출력 Channel.

- CONFIG_UART_RX_INTERRUPT
  이 부분이 정의되면 uart.c에서 수신 Interrupt를 처리 하기 위해 Ring_buffer가 정의되고 함수에서 수신 Interrupt를 처리할 수 있도록 Compile 한다.

  정의되지 않으면 Ring_buffer를 사용하지 않고 UART Channel Receiver Buffer Register만 사용 하도록 Compile 한다.

사용자가 이 구성을 따르지 않는 UART Application을 제작할 경우 위의 정의부분도 수정하면 된다.

4 Point This Note

- CANTUS의 UART Register에 관한 설정은 UartConfig() 함수를 호출함으로써 이뤄진다.
- CANTUS의 UART 설정 시 ＼Cantuslib＼Cantus_lib_config.h를 확인하고,
  필요할 경우 수정하여야 한다.
- CANTUS의 UART에 관한 내용은 CANTUS Datasheet 13 UART를 참고하라.