Application Note

I2C_MasterSlave

Arm and Keil are registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

All other company names, product names, and service names mentioned herein may be trademarks of their respective companies.

2022-04-08
Rev 1.0

Table of Contents

1. Preface

This application note describes sample software for the I2C Master and Slave control functions using the I2C driver.
This document helps the user check operation of a product under development and develop its program.

2. Technical Term

Term/Abbreviation	Definition
I2C	Inter-Integrated Circuit
BSP	Board Support Package
UART	Universal Asynchronous Receiver Transmitter

3. Reference Document

| Document	Notes
Data sheet	Refer to the data sheet of MCU to be used.
Reference manual	Refer to the reference manual of each IP to be used.
Application note MCU User Guide	Refer to the MCU user manual to be used.
Driver API list	Refer to the MCU Doc folder to be used.

4. Target Sample Program

Sample Program	Outline
I2C_MasterSlave	Sample program of I2C_MasterSlave function

5. Configuration Diagram
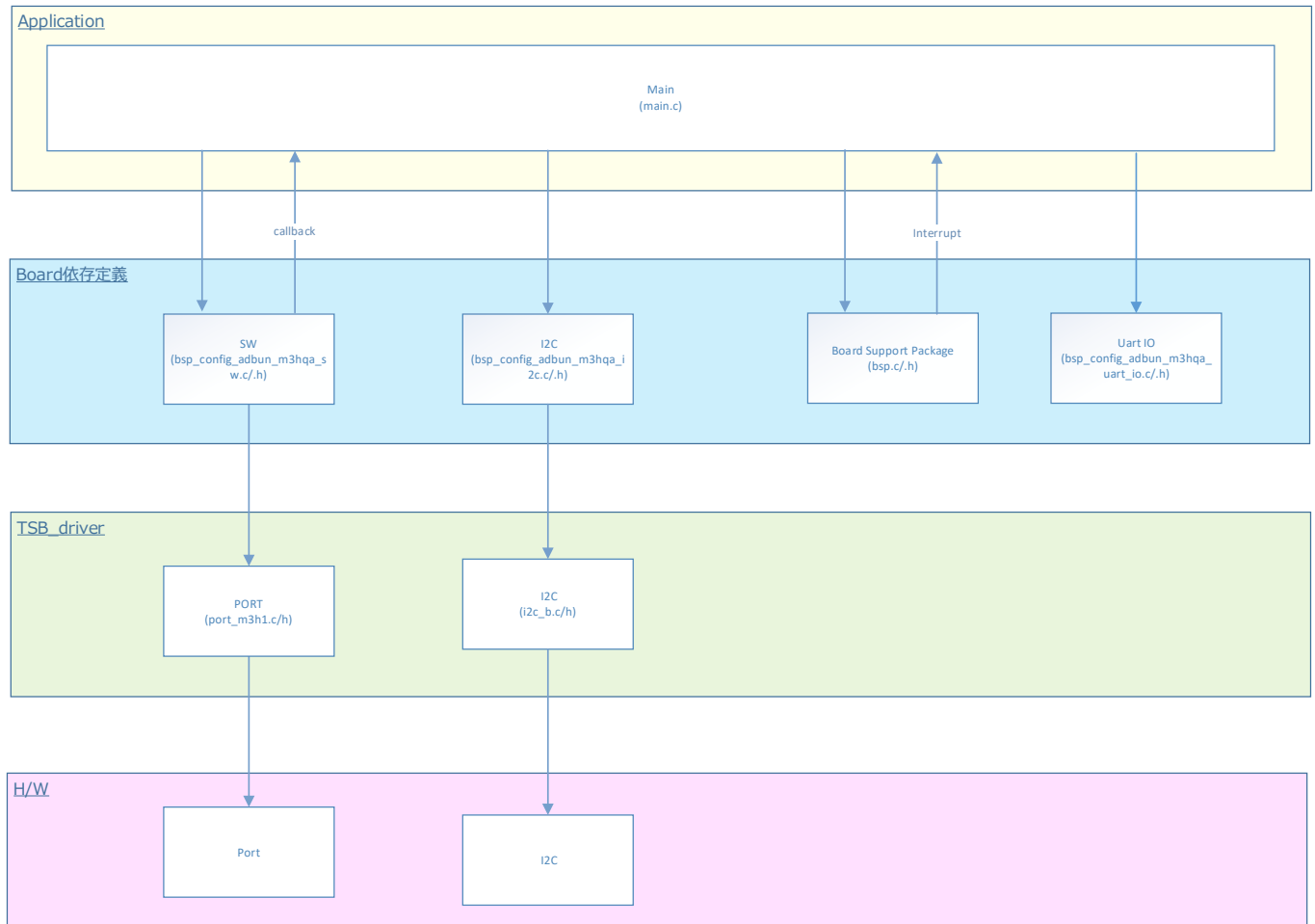
6. Sample Program：I2C_MasterSlave

This is sample software to check the operation of EI2C Master and Slave according to the command input from the terminal emulator.

6.1. Outlines of Operation

It is divided into Master operation and Slave operation.
Master: Enter commands (write, read) from the terminal emulator.
Slave: When a Request from Master is received, I2C control is used and the result is output according to the command.
Slave_address is displayed in the terminal emulator as "sa xx".

Command list

Command	function	Parameters (hex)		Input sample	Note
		1	2		
write	Data transmission	XX	XX…	"write" "write 60" "write 6011223344"	・[dataA_form1]: Send INIT_SLAVE with no parameters ・[dataA_form2]: Send INIT_WDATA with no parameters ・The transmission size is the total number of bytes of [dataA_form1] + [dataA_form2]
		[dataA_form1] (Slave_address) (1 byte)	[dataA_form2] (Master_tx_data) (max 4 byte)		
read	Data transmission + Data reception	XX	XX…	"read" "read 60" "read 600001"	・[dataB_form1]: Same specifications as [dataA_form1] ・[dataB_form2]: This is the read size when reading from the Sub Address or later of [dataB_form1]. Send INIT_RNUM with no parameters ・Send / receive size Send: Number of bytes in [dataB_form1] Receive: Number of bytes in [dataB_form2]
		[dataB_form1] (Slave_address) (1 byte)	[dataB_form2] (Master_tx_data) (1 byte, 2 byte)		
slave	Switch to Slave Mode	-	-	"slave"	・[dataC_form1]: Same specifications as [dataA_form1] ・After switching, wait for the Slave Address to be received. ・If you make a request more than the number of times set in SLAVE_CONTINUE, it will return from Slave mode to Master mode
		-	-		

Note 1）"XX" is a hex number. For 0x12, enter "12".
Note 2）The parameter of [ ] is accepted even if it is not specified.

6.2. Function to Use

The functions to use are as follows.
For the Port assignment of each channel, refer to the MCU user manual.

IP	Channel	Objective
I2C	BSP_ I2C_1	For I2C control. Works as a Master Device or Slave Device
UART	BSP_UART_1	For terminal emulator communication. Used for operation log output and command input

6.3. Interrupt to Use

Interrupt	Outlines
INTI2C1NST	I2C ch.1 Status interrupt
INTUART0RX	UART ch0 Receive interrupt. For terminal emulator
INTUART0TX	UART ch0 Transmission interrupt. For terminal emulator
INTUART0ERR	UART ch0 Error interrupt. For terminal emulator

6.4. Configuration

"main.c" configuration setting.

Configuration	Current Value	Description
INIT_SLAVE	Slave Address	-
INIT_WDATA	11, 22, 33, 44	4 bytes are used as write data
INIT_RNUM	0x02	Number of read request bytes of 2 bytes
Sub address	Start sub Address	-
PARAM A Size Max	0x04	Number of received bytes of [dataA_form2]
PARAM B Size Max	0x02	Number of received bytes of [dataB_form2]
SLAVE_CONTINUE	10	When Slave is operating, if you make a request that exceeds the number of times set in SLAVE_CONTINUE, it will return from Slave mode to Master mode

6.5. Example of Terminal Emulator Output
6.5.1. Normal Operation
Write Command

Master   **Slave (slave address C0)**

When [dataA_form1] and [dataA_form2] are saved

```
command > write
master
sa   60
tx[0] 11
tx[1] 22
tx[2] 33
tx[3] 44
```

write command →

INIT_SLAVE
=0x60 (case)

INIT_WDATA
=11,22,33,44 (case)

```
rx[0] 11
rx[1] 22
rx[2] 33
rx[3] 44
slave
sa   60
```
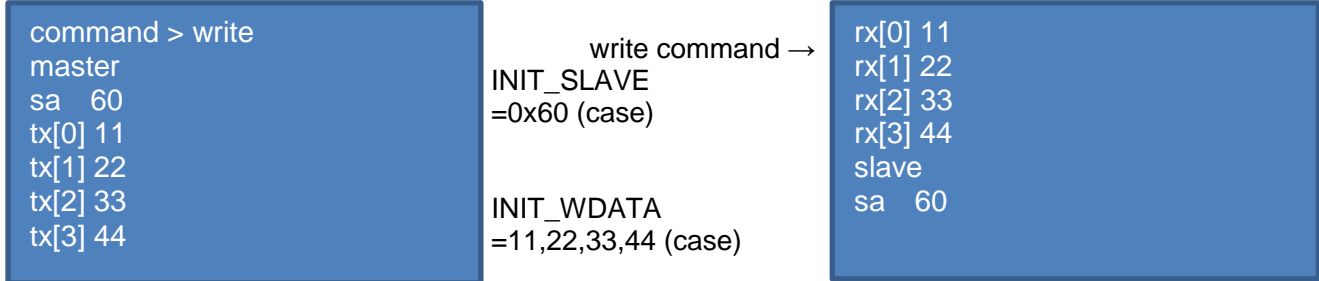
When [dataA_form2] is saved

```
command > write 60
master
sa   60
tx[0] 11
tx[1] 22
tx[2] 33
tx[3] 44
```

write command →

INIT_WDATA
=11,22,33,44 (case)

```
rx[0] 11
rx[1] 22
rx[2] 33
rx[3] 44
slave
sa   60
```
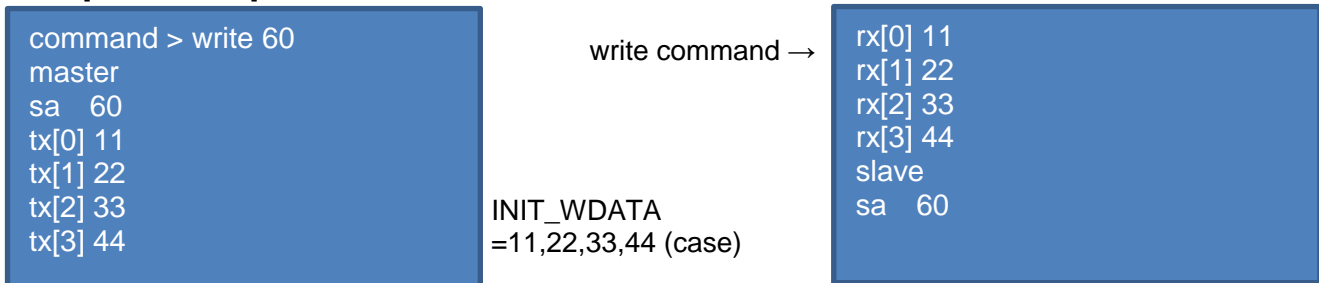
When [dataA_form1] and [dataA_form2] are not labor-saving

```
command > write 6055667788
master
sa   60
tx[0] 55
tx[1] 66
tx[2] 77
tx[3] 88
```

write command →

```
rx[0] 55
rx[1] 66
rx[2] 77
rx[3] 88
slave
sa   60
```

2022-04-08
Rev 1.0
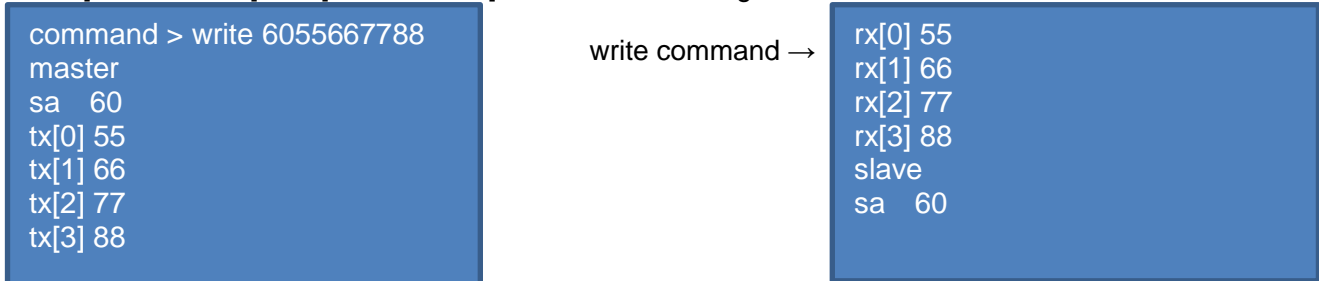
Read Command

Master		Slave (slave address C0)

When [dataB_form1] and [dataB_form2] are saved

```
command > read
master
sa   60
rx[0] 11
rx[1] 22
```

read command →

INIT_SLAVE
=0x60 (case)

INIT_RNUM
=0x0002 (case)

```
tx[0] 11
tx[1] 22
slave
sa   60
```

When [dataB_form2] is saved

```
command > read 60
master
sa   60
rx[0] 11
rx[1] 22
```

read command →

INIT_RNUM
=0x0002 (case)

```
tx[0] 11
tx[1] 22
slave
sa   60
```

[When [dataB_form1] and [dataB_form2] are not labor-saving

```
command > read 600001
master
sa   60
rx[0] 11
```

read command →

```
tx[0] 11
slave
sa   60
```

2022-04-08
Rev 1.0

Slave Command

Master

```
-------------------
| I2C master mode |
-------------------
command >
```

slave command →

Slave

```
-------------------
| I2C master mode |
-------------------
command > slave
-------------------
| I2C slave mode |
-------------------
slave
sa   60
```

6.5.2. Case of Error Occurrence

Nothing.

7. I2C Driver

7.1. List of driver

The I2C is controlled by using the following interface.
For an example of use, refer to the source code.

Driver	Control Outlines
I2C_init	I2C Register initialization
I2C_start_condition	Generate start condition
I2C_get_clock_setting	Return I2C clock settings
I2C_slave_init	Slave mode setting

7.2. Details

See "3. Reference Documents" for more information.

8. Revision History

Revision	Date	Description
1.0	2022-04-08	First release

2022-04-08
Rev 1.0

RESTRICTIONS ON PRODUCT USE

Toshiba Electronic Devices & Storage Corporation

https://toshiba.semicon-storage.com/