

アプリケーションノート

EI2C_MasterSlave

Arm および Keil は、Arm Limited（またはその子会社）の米国およびその他の国における登録商標です。

この資料に記載されている社名・商品名・サービス名などは、それぞれ各社が商標として使用している場合があります。

目次

| | |
|---------------------------------------|----|
| 目次 | 2 |
| 1. はじめに..... | 3 |
| 2. 用語..... | 3 |
| 3. 関連するドキュメント..... | 3 |
| 4. 対象サンプルプログラム..... | 4 |
| 5. 構成図 | 4 |
| 6. サンプルプログラム : EI2C_MasterSlave | 5 |
| 6.1. 動作・操作概要..... | 5 |
| 6.2. 使用する機能..... | 6 |
| 6.3. 使用する割り込み | 6 |
| 6.4. コンフィグレーション | 6 |
| 6.5. ターミナルソフト出力例 | 7 |
| 6.5.1. 正常時 | 7 |
| 6.5.2. エラー発生時 | 9 |
| 7. EI2C Driver | 9 |
| 7.1. 一覧 | 9 |
| 7.2. Driver 詳細 | 9 |
| 8. 改訂履歴..... | 10 |
| 製品取り扱い上のお願ひ..... | 11 |

1. はじめに

本書は、EI2C ドライバーを用いた、EI2C Master および Slave 制御機能を用いる製品を開発する際、動作確認用または、プログラム開発の参考としてご利用願います。

2. 用語

| 用語/略語 | 定義 |
|-------|--|
| EI2C | Inter-Integrated Circuit interface version A |
| BSP | Board Support Package |
| UART | Universal Asynchronous Receiver Transmitter |

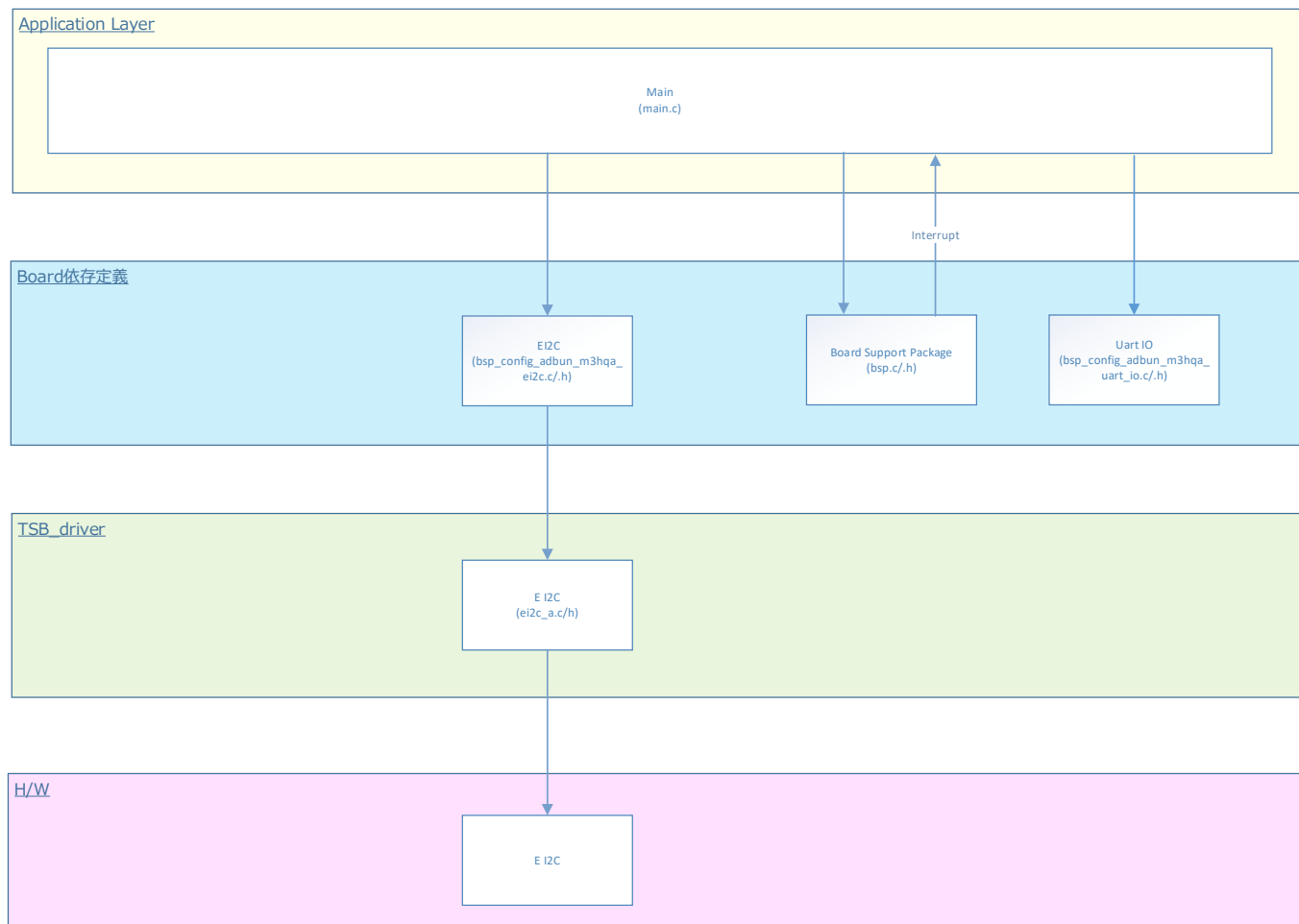
3. 関連するドキュメント

| ドキュメント | 備考 |
|--------------------------|--------------------------------|
| データシート | 利用する MCU のデータシートを参照してください |
| リファレンスマニュアル | 利用する各 IP のリファレンスマニュアルを参照してください |
| アプリケーションノート MCU 利用説明書 | 利用する MCU 利用説明書を参照してください |
| Driver API 一覧 | Doc フォルダを参照してください |

4. 対象サンプルプログラム

| サンプルプログラム | 概要 |
|------------------|-------------------------------|
| EI2C_MasterSlave | EI2C_MasterSlave 機能のサンプルプログラム |

5. 構成図



6. サンプルプログラム : EI2C_MasterSlave

ターミナルソフトから入力されたコマンドに従って、EI2C Master および Slave 時の動作を確認するサンプルソフトです。

6.1. 動作・操作概要

Master 動作、Slave 操作に分かれます。

Master : ターミナルソフトからコマンド(書き込み、読み込み)を入力します。

Slave : Master からの Request を受けたら、EI2C 制御を使用し、コマンドに従って結果を出力します。

Slave_address は、"sa xx"としてターミナルソフトに表示されます。

コマンド一覧

| コマンド | 説明 | パラメータ(hex) | | 入力例 | 備考 |
|-------|-----------------------|--|---|---|---|
| | | 1 | 2 | | |
| write | データ送信 | XX | XX... | "write" "write 60" "write 6011223344" | <ul style="list-style-type: none"> • [dataA_form1]: パラメータ無しの場合 は INIT_SLAVE を送信 • [dataA_form2]: パラメータ無しの場合 は INIT_WDATA を送信 • 送信サイズは [dataA_form1] + [dataA_form2]の合計バ イト数 |
| | | [dataA_form1] (Slave_address) (1 byte) | [dataA_form2] (Master_tx_data) (max 4 byte) | | |
| read | データ送信 + データ受信 | XX | XX... | "read" "read 60" "read 600001" | <ul style="list-style-type: none"> • [dataB_form1]: [dataA_form1]と同一 仕様 • [dataB_form2]: [dataB_form1]の Sub Address 以降から読む 際の read size です。 パラメータ無しの場合 は INIT_RNUM を送信 • 送受信サイズは 送信 : [dataB_form1] のバイト数 受信 : [dataB_form2] のバイト数 |
| | | [dataB_form1] (Slave_address) (1 byte) | [dataB_form2] (Master_tx_data) (1 byte, 2 byte) | | |
| slave | Slave Mode へ の切り替え | - | - | "slave" | <ul style="list-style-type: none"> • [dataC_form1]: [dataA_form1]と同一 仕様 • 切り替え後は Slave Address の受信を待ち ます。 • SLAVE_CONTINUE で 設定した回数を超える Request を行うと Slave mode から Master mode へ戻りま す。 |
| | | - | - | | |

注 1) "XX"は 16 進数。0x12 の場合は"12"と入力する。

注 2) []のパラメータは指定なしでも受け付ける。

6.2. 使用する機能

使用する機能は下記のとおりです。

BSP Channel の Port 割り当ては、MCU 利用説明書を参照願います。

| IP | チャンネル | 目的 |
|------|------------|--|
| EI2C | BSP_EI2C_1 | EI2C 制御用。Master Device または Slave Device として動作します |
| UART | BSP_UART_1 | ターミナルソフト出力用。動作 Log 出力、およびコマンド入力に使用します |

6.3. 使用する割り込み

| 割り込み | 制御概要 |
|-------------|---------------------------|
| INTI2C1NST | EI2C ch.1 ステータス割り込み |
| INTI2C1ATX | EI2C ch.1 送信バッファエンプティ割り込み |
| INTI2C1BRX | EI2C ch.1 受信バッファエンプティ割り込み |
| INTUART0RX | UART ch0 受信割り込み。端末通信用 |
| INTUART0TX | UART ch0 送信割り込み。端末通信用 |
| INTUART0ERR | UART ch0 エラー割り込み。端末通信用 |

6.4. コンフィグレーション

main.c コンフィグレーション設定

| コンフィグレーション | 設定値 | 説明 |
|------------------|-------------------|---|
| INIT_SLAVE | Slave Address | - |
| INIT_WDATA | 11, 22, 33, 44 | 4byte を書き込みデータとします |
| INIT_RNUM | 0x02 | 2byte の Read 要求 byte 数です |
| Sub address | Start sub Address | - |
| PARAM A Size Max | 0x04 | [dataA form2]の受付 byte 数です |
| PARAM B Size Max | 0x02 | [dataB form2]の受付 byte 数です |
| SLAVE_CONTINUE | 10 | Slave 動作時、SLAVE_CONTINUE で設定した回数を超える Request を行うと Slave mode から Master mode へ戻ります |

6.5. ターミナルソフト出力例

6.5.1. 正常時

Write Command

Master 例

Slave 例(slave address C0)

[dataA_form1], [dataA_form2]を省力した場合

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

write command 対応 →
INIT_SLAVE
=0x60 の場合

INIT_WDATA
=11,22,33,44 の場合

```
rx[0] 11
rx[1] 22
rx[2] 33
rx[3] 44
slave
sa 60
```

[dataA_form2]を省力した場合

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

write command 対応 →

INIT_WDATA
=11,22,33,44 の場合

```
rx[0] 11
rx[1] 22
rx[2] 33
rx[3] 44
slave
sa 60
```

[dataA_form1], [dataA_form2]を未省力の場合

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

write command 対応 →

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

Read Command

Master 例

[dataB_form1], [dataB_form2]を省力した場合

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

read command 対応 →

INIT_SLAVE
=0x60 の場合

INIT_RNUM
=0x0002 の場合

Slave 例(slave address C0)

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

[dataB_form2]を省力した場合

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

read command 対応 →

INIT_RNUM
=0x0002 の場合

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

[dataB_form1], [dataB_form2]を未省力の場合

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

read command 対応 →

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


Slave Command

Master 例

```

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

slave command 対応 →

Slave 例

```

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

6.5.2. エラー発生時

特に無し。

7. EI2C Driver

7.1. 一覧

下記 Driver を使用して、EI2C を制御します。
使用例はソースコードを参照してください。

| Driver 名 | 制御概要 |
|-----------------------|---------------|
| EI2C_init | EI2C レジスタの初期化 |
| EI2C_restartcondition | 再起動条件を生成 |
| EI2C_startcondition | 起動条件を生成 |
| EI2C_slave_init | スレーブモード設定 |

7.2. Driver 詳細

Driver の詳細に関しては、「3. 関連するドキュメント」を参照してください。

8. 改訂履歴

| Revision | 日付 | 変更項目 |
|----------|------------|------|
| 1.0 | 2022-04-08 | 初版 |

