

# UARTコマンド仕様書

---

項目	内容
作成者	starfort@nifty.com
レビジョン	00.01.01
発行日	2016/12/10
来歴	00.01.00 : 2016/10/18 : 新規作成
	00.01.01 : 2016/12/10 : Slave Address関連の関数を追加

---

## 1. 目的

本仕様書では、I2C接続されたスレーブデバイス内部レジスタへ書き込みを行うツールのコマンドを規定する。  
このツールは NXP 製マイクロコントローラ LPC-11U37 用ソフトウェアのソースコードとして提供される。  
動作条件等については、当該ツールの仕様書を参照のこと。

---

## 2. コマンド・リファレンス

---

### 1) put\_slvaddr

- 書式 : put\_slvaddr [slave\_address]
- 値 : 設定値: slave\_address - スレーブ・アドレス ( 7-bit : 0x00 ~ 0x7f )  
戻り値: 成功時 = 正常終了メッセージ  
失敗時 = エラー・メッセージ
- 説明 : 読み書きする I2C デバイスのスレーブ・アドレスを設定する。
- ノート: スレーブ・アドレスの表記は 16 進数 (小文字を使用) です。  
値の範囲は 7bit で LSB 詰めです。(0x00 ~ 0x7f)  
10bit 拡張アドレスには対応しておりません。
- 例 :

```
>put_slvaddr 0x1e
#exit successfully
>
>put_slvaddr 0xfe
#error : illegal parameter
>
```

## 2) get\_slvaddr

- 書式 : get\_slvaddr
- 値 : 設定値:なし  
戻り値:成功時=設定されているアドレス  
失敗時=エラー・メッセージ
- 説明 : 設定されているI2Cデバイスのスレーブ・アドレスを取得する。
- ノート: スレーブ・アドレスは 7bit で LSB 詰めです。(0x00 ~ 0x7f)  
10bit拡張アドレスには対応していません。
- 例 :

```
>get_slvaddr
#slave address : 0x1e
>
>get_slvaddr
#error : not defined
>
```

## 3) put\_reg

- 書式 : put\_reg [resister\_address] [resister\_data]
- 値 : 設定値:resister\_address - 書き込むI2Cデバイス・レジスタのアドレス( 16-bit : 0x0000 ~ 0xFFFF )  
resister\_data - I2Cデバイス・レジスタへ書き込むデータ( 8-bit : 0x00 ~ 0xFF )  
戻り値:成功時=正常終了メッセージ  
失敗時=エラー・メッセージ
- 説明 : スレーブ接続されたI2Cデバイス内部のレジスタへ 1 Byteデータを書き込む。
- ノート: I2Cデバイスのスレーブ・アドレスを事前に設定しておく必要があります。  
スレーブ・アドレスを同時に設定する場合は put\_regEXコマンドを使用して下さい。
- 例 :

```
>put_reg 0x01ac 0x5a
#exit successfully
>
>put_reg 0x01ac 0x5a3c
#error : illegal parameter
>
```

## 4) get\_reg

- 書式 : `get_reg [resister_address]`
- 値 : 設定値: `resister_address` - 読み込むI2Cデバイス・レジスタのアドレス( 16-bit : 0x0000 ~ 0xFFFF )  
戻り値: 成功時=読み込んだI2Cデバイス・レジスタのデータ  
失敗時=エラー・メッセージ
- 説明 : スレーブ接続されたI2Cデバイス内部のレジスタから 1 Byteデータを読み込む。
- ノート: I2Cデバイスのスレーブ・アドレスを事前に設定しておく必要があります。  
スレーブ・アドレスを同時に設定する場合は `get_regEX`コマンドを使用して下さい。
- 例 :

```
>get_reg 0x01ac
#0x5a
>
>get_reg 0x1a
#error : illegal parameter
>
```

## 5) put\_regEX

- 書式 : `put_regEX [slave_address] [resister_address] [resister_data]`
- 値 : 設定値: `slave_address` - スレーブ・アドレス( 7-bit : 0x00 ~ 0x7f )  
`resister_address` - 書き込むI2Cデバイス・レジスタのアドレス( 16-bit : 0x0000 ~ 0xFFFF )  
`resister_data` - I2Cデバイス・レジスタへ書き込むデータ( 8-bit : 0x00 ~ 0xFF )  
戻り値: 成功時=正常終了メッセージ  
失敗時=エラー・メッセージ
- 説明 : スレーブ接続されたI2Cデバイス内部のレジスタへ 1 Byteデータを書き込む。
- ノート: スレーブ・アドレスを設定しない場合は `put_reg`コマンドを使用して下さい。
- 例 :

```
>put_regEX 0x3c 0x01ac 0x5a
#slave address : 0x3c
#exit successfully
>
>put_regEX 0xfc 0x01ac 0x5a
#error : illegal parameter
>
```

## 6) get\_regEX

- 書式 : `get_regEX [slave_address] [resister_address]`

- 値 : 設定値: slave\_address - スレーブ・アドレス ( 7-bit : 0x00 ~ 0x7f )  
resister\_address - 読み込むI2Cデバイス・レジスタのアドレス ( 16-bit : 0x0000 ~ 0xFFFF )  
戻り値: 成功時 = 読み込んだI2Cデバイス・レジスタのデータ  
失敗時 = エラー・メッセージ
- 説明 : スレーブ接続されたI2Cデバイス内部のレジスタから 1 Byteデータを読み込む。
- ノート: スレーブ・アドレスを設定しない場合は get\_regコマンドを使用して下さい。
- 例 :

```
>get_regEX 0x3c 0x01ac
#0x5a
>
>get_regEX 0xfc 0x01ac
#error : illegal parameter
>
```

---

## 3. 特記事項

---

### 1) シリアルポート設定

シリアルポートの設定は下記に従って下さい。

item	value
baud rate	115200(bps)
data bits	8(bit)
stop bit	1(bit)
parity	none
X-On char	0x11
X-Off char	0x13
RTS	disable
DTS	disable

### 2) ターミナル画面入力

- コマンド行の最大文字数は63文字です。(以降は無視)
  - 改行コードは CR+LF として下さい。
  - コマンドおよびパラメータのデリミタは、1つ以上の空白文字(半角スペース)です。
-