REX-USB61mk2
I2Cレジスタアクセスツール

ユーザーズマニュアル

2016 年 8 月

第 1.0 版

ラトックシステム株式会社

第1章 レジスタアクセスツールについて

本ツールでは、REX-USB61mk2 に接続した I2C デバイスのレジスタ情報の確認と変更を行うことができ、以下の特徴があります。

- Windows アプリケーションの GUI 画面にて、容易に I2C デバイスの各レジスタ値の読み出しや設定が可能。
- I2C デバイスの仕様書等にあるレジスタフィールドの表記と同様な表示が可能。
- I2C デバイスのレジスタ表示詳細を XML で記述が可能。
- 読み出したレジスタ値の CSV ファイル保存が可能。

具体的には、以下の手順で使用します。(EPSON RealTimeClock RTC-8564 での例)

① I2C デバイスの仕様を確認

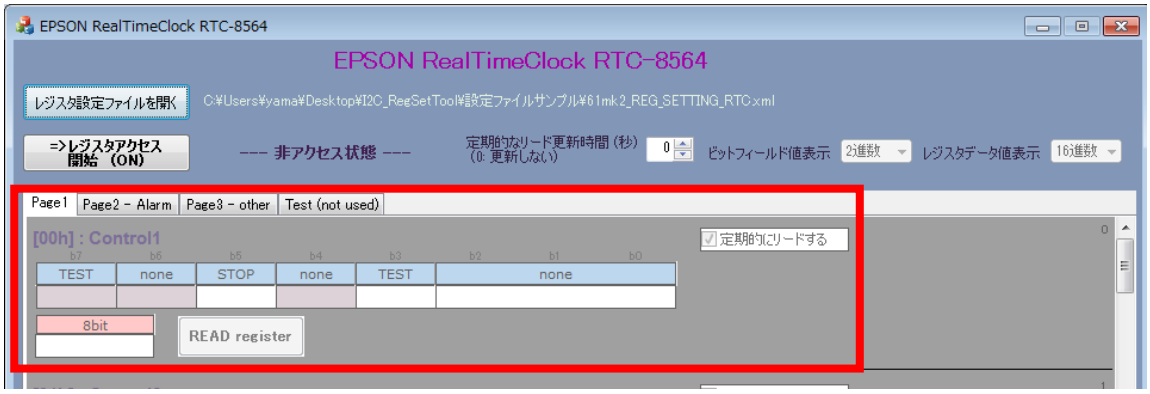
Address [h]	Function	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
00	Control 1	TEST	0	STOP	0	TEST	0	0	0

② 本ツール用の XML ファイルを作成（以下の内容はサンプルの抜粋）

```
<REGISTER NAME="Control1" REGADDRESS="0x00" SIZE="8bit">
  <IntervalRead> YES </IntervalRead>
  <BITFIELD NAME="none" bitSize="3" ACCESSTYPE="RW" />
  <BITFIELD NAME="TEST" bitSize="1" ACCESSTYPE="RW" />
  <BITFIELD NAME="none" bitSize="1" ACCESSTYPE="R" />
  <BITFIELD NAME="STOP" bitSize="1" ACCESSTYPE="RW" />
  <BITFIELD NAME="none" bitSize="1" ACCESSTYPE="R" />
  <BITFIELD NAME="TEST" bitSize="1" ACCESSTYPE="R" />
```

③ 本ツールに XML ファイルを読み込ませる

ビット単位でのアクセスを行うことにより、より詳細なデバイスの評価が可能となります。



(1-1) 動作環境

対応 OS：Windows10 / 8.1 / 7 / Vista
CPU：マルチコア CPU
メモリ：使用可能 RAM 1GB 以上
画面解像度：1248 × 1024 以上の画面サイズ
対応デバイス：I2C スレーブデバイスで、レジスタ(メモリ空間)には以下のプロトコルでアクセス可能であること。

■ サポートする I2C プロトコルとその定義

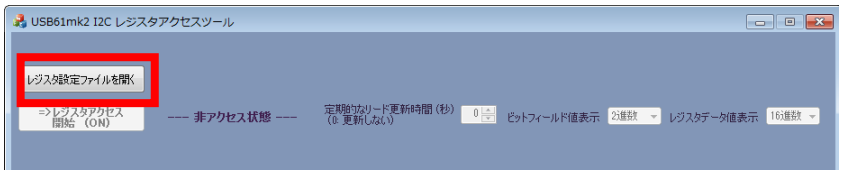
定義	意味
[S]	Start
[Sr]	repeat Start
[P]	Stop
[SlaveAdrs7]	I2C スレーブアドレス（7 ビットアドレス）
[SlaveAdrs10-H]	I2C スレーブアドレス（10 ビットアドレスの上位ビット）
[SlaveAdrs10-L]	I2C スレーブアドレス（10 ビットアドレスの下位ビット）
<REG_ADRS>	レジスタアドレス（サブアドレス） アドレスサイズの指定によって 1 バイト～4 バイトが出力される。
<ReadData>	リードデータ
<WriteData>	ライトデータ
W	Write bit
R	Read bit
(ACK/NACK ビットは省略)	

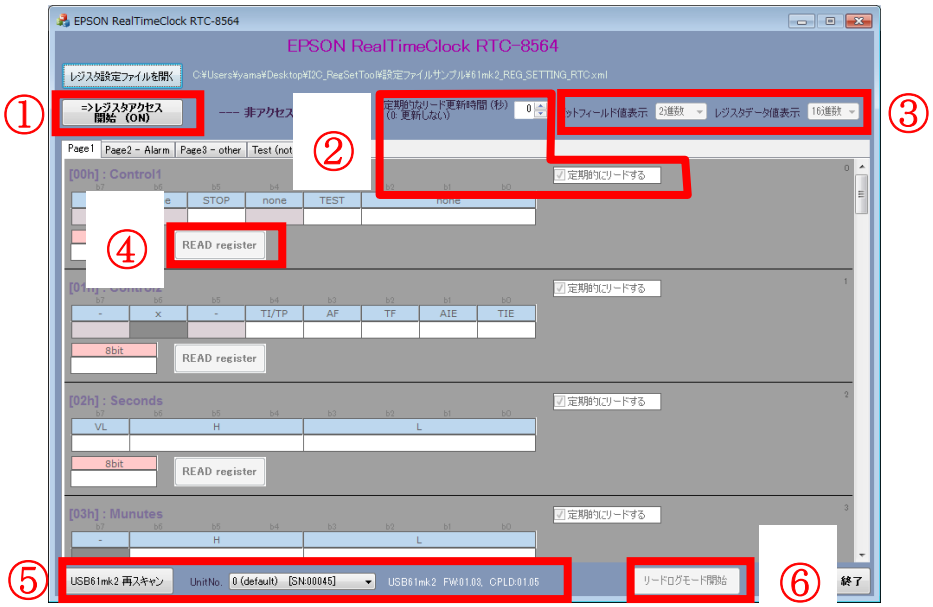
■ 対応デバイスのプロトコル

レジスタリード	
スレーブアドレス 7 ビット	[S] [SlaveAdrs7].W <REG_ADRS> [Sr] [SlaveAdrs7].R <ReadData> (<ReadData> ...) [P]
スレーブアドレス 10 ビット	[S] [SlaveAdrs10-H].W [SlaveAdrs10-L] <REG_ADRS> [Sr] [SlaveAdrs10-L].R <ReadData> (<ReadData> ...) [P]
レジスタライト	
スレーブアドレス 7 ビット	[S] [SlaveAdrs7].W <REG_ADRS> <WriteData> (<WriteData>...) [P]
スレーブアドレス 10 ビット	[S] [SlaveAdrs10-H].W [SlaveAdrs10-L] <REG_ADRS> <WriteData> (<WriteData>...) [P]

(1-2) レジスタアクセスツール概要

USB61mk2_REG.exe を起動すると、下記画面が表示されます。
「レジスタ設定ファイルを開く」をクリックし、作成した設定ファイルを開きます。
(以降はサンプルの 61mk2_REG_SETTING_RTC.xml を開いた場合の例となります。)





① -- 設定した内容でレジスタアクセスを開始(停止)します。
② -- チェックを入れると、設定した時間毎にリードを行い、表示内容を更新します。
③ -- ビットフィールド値/レジスタデータ値の表示方法を 2 進数/10 進数/16 進数から選択します。
④ -- 手動で各レジスタ値のリードを行い、表示内容を更新します。
また、ライト可能なフィールドデータ(グレー表示はリードオンリー)を書き換え[Enter]キーを押すとレジスタへの書き込みが行われます。
以下の記述方法の場合は、「③」の設定は無視されます。

入力データ	データの解釈	例
0xXX / XXh / XX.h / XX(h)	16 進数	0x5A
XX.b / XX(b)	2 進数	01101001.b
XX.d / XX(d)	10 進数	2531.d

(大文字・小文字は区別しない)

データ確定前に[ESC]キーを押すと変更前の値に戻ります。

⑤ -- 接続されている REX-USB61mk2 の情報を表示します。
⑥ -- 「リードログモード開始」をクリックすると、リードしたデータを CSV ファイル保存します。
(開始中はレジスタへのライトを行うことができません。)

CSV ファイル形式

リードしたレジスタを 1 行として、カンマ区切りで“ ”で囲んだ文字列として保存されます。

列番号	列のデータ	説明
1	時：分：秒	レジスタをリードした時刻(時分秒)。 ex：“16:13:03”
2	ミリ秒	レジスタをリードした時刻のミリ秒。 ex：“014”
3	レジスタアドレス	レジスタアドレスを 16 進数(末尾 h) で表記。アドレスサイズ(サブアドレス)長に応じて桁数が異なる。 ex：“1Ah”
4	レジスタ名	レジスタ名。 ex：”STATUS_REG”
5	レジスタ値	リードしたレジスタの値。 アプリケーションの「レジスタデータ値表示」で指定した形式で表記。 ex：“0xDEAD” -- 16 進数形式 “1101111010101101 (b)” -- 2 進数形式 “57005” -- 10 進数形式
6	ビットフィールド名 1	ビットフィールド名を表記。 （レジスタの上位ビットから） ex：“INT_ST”
7	ビットフィールド値 1	対応するビットフィールドの値を、アプリケーションの「ビットフィールド値表示」で指定した形式で表記。 ex：“01101(b)” -- 2 進数形式 “0xD” -- 16 進数形式 “13” -- 10 進数形式
…	… 以下、レジスタで定義したビットフィールド名とビットフィールド値が続く。 順序は上位ビットから下位ビットの順となる。	

(1-3) レジスタ設定ファイルについて

レジスタ設定ファイルの定義と注意点

・レジスタ設定ファイルは、XML で記述します。
・エンコード形式は UTF-8/UTF-16 と Shift-JIS に対応しています。
・XML 宣言文として以下のいずれかが設定ファイルの先頭に必要で、UTF-16 の場合はファイル先頭（XML 宣言文の前）に 2 バイトの BOM コードも必要です。

XML 宣言文	エンコード形式
<?xml version="1.0" encoding="UTF-8"?>	UTF-8
<?xml version="1.0" encoding="Shift-JIS"?>	S-JIS
<?xml version="1.0" encoding="UTF-16"?>	UTF-16 （BOM コード必須） BOM コード 0xFE 0xFF：UTF-16BE（ビックエンディアン） 0xFF 0xFE：UTF-16LE（リトルエンディアン）
<?xml version="1.0"?>	UTF-8 (encoding を指定しなかった場合のデフォルト)

・XML の仕様では、大文字と小文字を区別しますが、本ツールでは大文字と小文字を区別しません。
・１６進数を記述する場合、値の先頭に 0x を付けるか、値の末尾に h を付けることで１６進数と判断します。
ex1: 0x23
ex2: 23h
・10 進数の数値を記述する際に、値の先頭に 0 を記述しないでください。(この場合 8 進数と判断されます。)
ex: 014 ・・・NG（12 と解釈されます。）
・アドレス値(I2C スレーブアドレス、レジスタアドレス)は、16 進数で記述してください。

タグ仕様

root

唯一のルート要素

記述	< root > ……. </root>
アトリビュート	なし
設定値	－
省略時の値	省略不可
子要素	config, controller, body
親	－（唯一のルート）
使用例	<root> <config> ….. </config> <controller> ….. </controller> <body> …. </body> </root>

config

REX-USB61mk2 の設定情報をまとめた親要素

記述	<config> …… </config>
アトリビュート	なし
設定値	－
省略時の値	省略不可
子要素	mode, frequency, frequensy_HS, power, pullup, RegisterReadTime, ByteInterval
親	root
使用例	<config> <mode> I2c </mode> <frequency> 400 </frequency> …. </config>

controller

コントローラー名、アドレッシングサイズを定義する親要素

記述	< controller　description=”example controller” >　….　</controller>
アトリビュート	description = ”コントローラー名称”
設定値	－
省略時の値	省略不可
子要素	RegAddress_size
親	root
使用例	<controller　description=“CXP OpticalModule”> 　…. </controller> desctiption で定義した名称は、アプリケーション画面にテキスト表示する。

RegAddress_Size

デフォルトとなるレジスタアドレスのアドレス幅を指定する。I2C サブアドレスのバイト数となる。

記述	< RegAddress_Size > 数値または文字列 < /RegAddress_Size >
アトリビュート	なし
設定値	(数値)レジスタアドレスのサイズ: 1, 2, 3, 4 または 8bit, 16bit, 24bit, 32bit, BYTE, WORD, DWORD
省略時の値	－
子要素	－
親	controller
使用例	< RegAddress_Size > 16bit < /RegAddress_Size > デフォルトのレジスタアドレス幅は 16bit（レジスタのアドレス空間 0-FFFFh、I2C サブアドレスとして 2 バイトを使用）であることを定義する。

body

レジスタ設定情報をまとめた親要素

記述	<body> ……. </body>
アトリビュート	なし
設定値	－
省略時の値	省略不可
親	root
子要素	ByteOrder, TabPage
使用例	<body> <TabPage NAME = “Page1” > …. </tabPage> <TabPage NAME = “Page2” > …. </tabPage> <TabPage NAME = “Page3” > …. </tabPage> … </body>

mode

REX-USB61mk2 の動作モードを指定する。I2C のみ指定可能。本指定は省略可能。

記述	<mode> I2C </mode>
アトリビュート	なし
設定値	(文字列) I2C
省略時の値	I2C
子要素	－
親	config
使用例	<mode> I2C </mode>

frequency

周波数を指定する。1kHz 単位で指定する。

記述	<frequency>　周波数 </frequency>
アトリビュート	なし
設定値	(数値) 1～5000
省略時の値	400
子要素	－
親	config
使用例	< frequency > 200 < /frequency > 200kHz を指定する。

frequency_HS

HighSpeed モードでレジスタアクセスする場合の周波数を指定する。1kHz 単位で指定する。

記述	<frequency_HS>　周波数 </frequency_HS>
アトリビュート	なし
設定値	(数値) 1～5000
省略時の値	1000
子要素	－
親	config
使用例	< frequency_HS > 2500 < /frequency_HS > HS-mode アクセス時は、2500kHz(2.5MHz)を指定する。

power

REX-USB61mk2 の出力電源を指定する。

記述	<power> 数値または文字列 </power>
アトリビュート	なし
設定値	(数値または文字列) 0 または OFF ： 出力しない(OFF) 1, ON1 または 1.8V ：1.8V を出力 2, ON2 または 2.5V ：2.5V を出力 3, ON3 または 3.3V ：3.3V を出力 5, ON5 または 5V ：5.0V を出力
省略時の値	－
子要素	－
親	config
使用例	<power> 3 </power> 3.3V の電源出力を指定する。

pullup

I2C の SDA, SCL 信号をプルアップするかどうかを指定する。

記述	<pullup> 文字列 </pullup>
アトリビュート	なし
設定値	(文字列) ON, OFF ON：プルアップする OFF：プルアップしない
省略時の値	OFF
子要素	－
親	config
使用例	<pullup> ON </pullup> SDA, SCL 信号のプルアップを行う。

ByteInterval

ライトアクセス時に、バイトごとに挿入するディレイ時間をマイクロ秒で指定する。

記述	<ByteInterval> 数値 </ByteInterval>
アトリビュート	なし
設定値	(数値) 0 , 12 ～ 65535 マイクロ秒単位で指定する。範囲外の値を指定した場合は 0 が指定されたことになる。
省略時の値	0
子要素	－
親	Config
使用例	<ByteInterval>12 </ByteInterval> ライトアクセス時、バイトごとに 12 マイクロ秒のディレイが入る。

RegisterReadTime

レジスタの定期的なリード更新時間の初期値を秒単位で指定する。

記述	<RegisterReadTime> 数値 または OFF </RegisterReadTime>
アトリビュート	なし
設定値	(数値) 0 ～ 10　または OFF 0 または OFF：全てのレジスタでリード更新は行わない。(デフォルト) 1～10：定期的なリードを有効にしているレジスタに対し、指定した秒間隔でレジスタの更新を行う。
省略時の値	0
子要素	－
親	Config
使用例	<RegisterReadTime>2 </RegisterReadTime> 定期的なリードを有効にしているレジスタに対し、2 秒間隔でレジスタのリード更新を行う。

ByteOrder

複数バイト長（16/24/32 ビット）レジスタのバイトオーダーのデフォルトを指定する。
タブページごとに指定することや、個別のレジスタに対して <REGISTER>のアトリビュートで指定することも可能。

タグ	<ByteOrder> 文字列 </ByteOrder>
アトリビュート	なし
設定値	<table><tr><th>文字列</th><th>意味</th></tr><tr><td>BigEndian Big MSB</td><td>ビックエンディアン (上位、下位の並び)</td></tr><tr><td>LittleEndian Little LSB</td><td>リトルエンディアン (下位、上位の並び)</td></tr></table>
省略時の値	指定しなかった場合のデフォルトは BigEndian となる。
子要素	－
親	Body
使用例	<ByteOrder> BigEndian </ByteOrder> デフォルトとして、16/24/32 ビット長のレジスタはビックエンディアンであることを指定する。

TabPage

ページ（タブページ）ごとの項目をまとめた要素。

アトリビュートで、表示するページの見出しを指定する。

１つ以上のページを指定する必要がある。省略はできない。

記述	<TagPage caption=”str”> ..... </TabPage>
アトリビュート	caption=”ページの見出し” ページタブに表示する文字列を指定する
設定値	－
省略時の値	省略不可
子要素	SlaveAddress, ByteOrder, REGISTER
親	Body
使用例	< TabPage caption=”Page1” > …. < /TabPage >

SlaveAddress

このタブページ内のレジスタをアクセスする際に使用する I2C スレーブアドレス(デバイスアドレス)を指定する。<REGISTER>タグのアトリビュートで、レジスタごと個別に指定することも可能である。

記述	<SlaveAddress> 16 進数値 </SlaveAddress>
アトリビュート	なし
設定値	(16 進数値)　I2C スレーブアドレス I2C スレーブアドレス(デバイスアドレス)を 16 進数で指定する。 2 桁の 16 進数で指定した場合は 7 ビットアドレスと解釈し、3 桁の 16 進数で指定した場合は 10 ビットアドレスと解釈する。 例 1:　0x51 または 51h　=>　7 ビットアドレスで 51h 例 2:　0x151 または 151h　=>　10 ビットアドレスで 151h
省略時の値	－
子要素	－
親	TabPage
使用例	< SlaveAddress > 0x30 < /SlaveAddress > スレーブアドレスとして 30h(7 ビットアドレス)を指定する

ByteOrder

タブページ内の複数バイト長(16/24/32 ビット)レジスタのバイトオーダーのデフォルトを指定する。
<REGISTER>タグのアトリビュートで、レジスタごとに指定することも可能である。

タグ	<ByteOrder> 文字列 </ByteOrder>
アトリビュート	なし
設定値	<table><tr><th>文字列</th><th>意味</th></tr><tr><td>BigEndian Big MSB</td><td>ビックエンディアン (上位、下位の並び)</td></tr><tr><td>LittleEndian Little LSB</td><td>リトルエンディアン (下位、上位の並び)</td></tr></table>
省略時の値	指定しなかった場合、デフォルトは BigEndian となる。
子要素	―
親	TabPage
使用例	<ByteOrder> BigEndian </ByteOrder> このタブページ上の、16/24/32 ビット長のレジスタはビックエンディアンであることを指定する。

REGISTER

ページ上に表示するレジスタを定義する。

記述	< REGISTER　NAME=”str”　REGADDRESS=”RegAdrs” SIZE=”size”　ByteOrder=”str”　HS_MODE=”Yes” SlaveAddress=”SlaveAdrs” > …. < /REGISTER >
アトリビュート	NAME=”レジスタ名” レジスタ名を指定する。 REGADDRESS=”レジスタアドレス” または “DIO” (REGADRS=, ADDRESS=, ADRS= 表記も可能) レジスタアドレスを 16 進数で指定する。 2 桁の 16 進数で指定した場合 アドレス幅 8 ビット(サブアドレス長 1 バイト)と解釈。 4 桁の 16 進数で指定した場合 アドレス幅 16 ビット(サブアドレス長 2 バイト)と解釈。 6 桁の 16 進数で指定した場合 アドレス幅 24 ビット(サブアドレス長 3 バイト)と解釈。

8 桁の 16 進数で指定した場合
アドレス幅 32 ビット(サブアドレス長 4 バイト)と解釈。

例
“0x51” または “51h” => サブアドレス長 1 バイト
“0x0051” または “0051h” => サブアドレス長 2 バイト
“0x000051” または “000051h” => サブアドレス長 3 バイト
“0x00000051” または “00000051h” => サブアドレス長 4 バイト

レジスタアドレスではなく ”DIO” と指定することで、DIO ポートを選択することができる。
”DIO” を指定した場合は、他のアトリビュート指定は無効となる。

SIZE=”レジスタのサイズ”
レジスタのサイズを指定する。
”8”, “16”, “24”, “32” が可能。
(”8bit”, “16bit”, “24bit”, “32bit”, “BYTE”, “WORD”, “DWORD” の指定も可能)
DIO 指定の場合は、この指定は無効となり 8bit サイズが指定される。

ByteOrder=”文字列”
このレジスタのサイズが複数バイト長(16/24/32 ビット長)の場合、バイトオーダーを指定する。本属性を指定しなかった場合はデフォルトの設定値が使用される。
DIO 指定の場合は、この指定は無効となる。

文字列	意味
“BigEndian” “Big” “MSB”	ビックエンディアン (上位、下位の並び)
“LittleEndian” “Little” “LSB”	リトルエンディアン (下位、上位の並び)

HS_MODE=”yes” または “no”
“yes”: このレジスタへのアクセスには、HS-mode を使用することを指定する。マスターコードが送信される。
“no”: このレジスタへのアクセスに対して、HS-mode は使用しない。(デフォルト)

	本属性を指定しなかった場合は、このレジスタに対して HS-mode は使用しない。 DIO 指定の場合は、この指定は無効となる。 SlaveAddress=”スレーブアドレス” I2C スレーブアドレス(デバイスアドレス)を 16 進数で指定する。 2 桁の 16 進数で指定した場合 7 ビットアドレスと解釈。 3 桁の 16 進数で指定した場合 10 ビットアドレスと解釈。 例 1： “0x51” または “51h” => 7 ビットアドレスで 51h 例 2： “0x051” または “051h” => 10 ビットアドレスで 051h 本属性を指定しなかった場合はデフォルトの設定値が使用される。 DIO 指定の場合は、この指定は無効である。
設定値	－
省略時の値	省略不可
子要素	MasterCode, IntervalRead, BITFIELD
親	TabPage
使用例	<REGISTER NAME=”statusREG1” ADDRESS=”0x000C” SIZE=”8”> … </REGISTER>

MasterCode

マスターコード（スレーブアドレスに先行して送信される 1 バイトのデータ）を指定する。
マスターコードを指定せずに HS-mode アクセスが行われる場合は、デフォルトのマスターコードとして 08H が送信される。

記述	< MasterCode > 数値 < /MasterCode >
アトリビュート	なし
設定値	数値 ： 1 バイトの値(マスターコード)
省略時の値	(HS-mode アクセス時のみ、0x08)
子要素	－
親	REGISTER
使用例	<MasterCode> 0x0A </MasterCode> マスターコードとして 0Ah を送信する。

IntervalRead

このレジスタに対して、初期状態で「定期的にリードする」チェック状態の有無を指定する。

タグ	<IntervalRead>　YES または NO　</IntervalRead>
アトリビュート	なし
設定値	(文字列) YES, NO
省略時の値	TRUE
子要素	－
親	REGISTER
使用例	<IntervalRead> YES </IntervalRead>

BITFIELD

ビットフィールドを定義する。

ビット位置は、出現順で下位ビットから上位ビットへと割り当てられる。

(C 言語のビットフィールド構造体定義と同じ)

タグ	<BITFIELD　NAME=”str”　bitSize=”n”　ACCESSTYPE=”xx”　/>
アトリビュート	NAME = ”ビットフィールド名” 　フィールドに表示する文字列を指定する。 bitSize = ”ビットサイズ” 　フィールドのサイズをビット数で指定する。 　省略時は 1 となる。 ACCESSTYPE = ”アクセスタイプ” 　“R”：ReadOnly,　“RW”：ReadWrite,　“N”：未使用 　“IN”：ReadOnly (“R”と同じ),　“OUT”：ReadWrite (“RW”と同じ) 　省略時は “RW” 指定となる。
設定値	－
省略時の値	－
子要素	なし
親	REGISTER

使用例

```
<REGISTER  NAME="TimerControl" REGADDRESS=”0x0E” SIZE = ”8bit” >
    <IntervalRead> TRUE </IntervalRead>
    <BITFIELD NAME="TDn" bitSize="2" ACCESSTYPE="RW" /> <!-- bit1,0 -->
    <BITFIELD NAME="－" bitSize="5" ACCESSTYPE="N" /> <!-- bit6-2 -->
    <BITFIELD NAME="TE" bitSize="1" ACCESSTYPE="R" /> <!—bit7 -->
</REGISTER>
```

上記例では、以下のようになる。

TimerControl：+0Eh

b7	b6	b5	b4	b3	b2	b1	b0
TE	－					TDn	

(1-4) サンプル設定ファイルについて

サンプル設定ファイル(61mk2_REG_SETTING_RTC.xml)の内容について説明いたします。(抜粋)


```
<?xml version="1.0" encoding="utf-8"?>
<root>
  <Config>
    <MODE>I2C</MODE>
    <FREQUENCY>400</FREQUENCY>
    <POWER>3</POWER>
    <PULLUP>ON</PULLUP>
  </Config>

  <Controller description="EPSON RealTimeClock RTC-8564">
    <RegAddress_Size> 1 </RegAddress_Size>
  </Controller>

  <Body>
    <TabPage caption="Page1">
      <SlaveAddress> 0x51 </SlaveAddress>

      <REGISTER NAME="Control1" REGADDRESS="0x00" SIZE="8bit">
        <IntervalRead> YES </IntervalRead>
        <BITFIELD NAME="none" bitSize="3" ACCESSTYPE="RW" />
        <BITFIELD NAME="TEST" bitSize="1" ACCESSTYPE="RW" />
        <BITFIELD NAME="none" bitSize="1" ACCESSTYPE="R" />
        <BITFIELD NAME="STOP" bitSize="1" ACCESSTYPE="RW" />
        <BITFIELD NAME="none" bitSize="1" ACCESSTYPE="R" />
        <BITFIELD NAME="TEST" bitSize="1" ACCESSTYPE="R" />
```

XML 宣言文

REX-USB61mk2 の設定

①に反映

②に反映

③に反映