

I2C センサー・ドライバー

ソフトウェア機能説明書

2017年3月 Ver0.1

株式会社 エフイーシー

お買い上げありがとうございます。

ご使用前に・・・・・・・・

- ・本説明書をよくお読みのうえ、正しくお使い下さい。
- ・必ず、「ソフトウェア使用許諾書」をお読みください。
- ・I2C 各デバイスについては各メーカー発行の Datasheet をお読みください。
- ・I2C 接続に関しては、I2C 各デバイスの Datasheet 及び
- ・お使いのワンボードマイコンの説明書をお読みください。

目次

| | |
|---------------------------------------|-----|
| 1.はじめにお読みください。 | 4 |
| 1-1.他社製品の登録商標および商標についてのおしらせ | 7 |
| 1-2.参考資料について | 8 |
| 1-3.I2C デバイス接続上の注意点 | 9 |
| 2.I2C とは・・ | 10 |
| 2-1.I2C センサードライバーとは・・ | 11 |
| 2-2.I2C センサードライバーの使い方 | 12 |
| 2-3.I2C センサー・ドライバーの使用上の注意 | 13 |
| 2-4.開発環境 | 14 |
| 2-5.本資料の使い方 | 15 |
| 3. 提供する I2C センサー・ドライバーの種類 | 16 |
| 3-1.12bit A/D コンバータ ADC121C021 | 17 |
| 3-2.デジタル光センサー TSL2561 | 40 |
| 3-3.3 軸ジャイロセンサー ITG3200 | 61 |
| 3-4.タッチセンサー MPR121 | 94 |
| 3-5.12bit D/A コンバータ M C P 4275 | 286 |
| 3-6.気温・湿度・気圧センサ BME280 | 302 |

| | |
|--------------------------------------|-----|
| 3-7.ステッピングモータコントローラ PCA9629A..... | 349 |
| 3-8.ジェスチャーセンサー PAJ7620U2..... | 427 |
| 3-9.液晶モジュールコントローラ PIC16F77A..... | 475 |
| 3-9-2. 液晶バックライト PCA9633..... | 499 |
| 3-10.16ch PWMコントローラ PCA9685 | 525 |
| 3-11.8×8マトリクスLEDコントローラ PCA9622 | 580 |
| 3-12.3軸加速度センサ ADXL345 | 633 |
| 4.最後に..... | 730 |

1.はじめにお読みください。

※※※ ご注意 ※※※

本書は、I2C インターフェースドライバーの解説書であり、I2C デバイスの各パラメータの説明書では有りません。

ご使用する際は、該当する I2C デバイスのデータシートに記載されている各レジスタと設定するパラメータと、その動作の関係を理解している事を前提に解説しています。

お客様へ：本ソフトウェアをご使用になる前に必ずソフトウェア使用許諾書をお読みください。

本ソフトウェアについては、「ソフトウェア使用許諾書」を承諾していただくことがご使用の条件となっています。あらかじめ内容をよくご確認の上、同意していただける場合のみご使用ください。

第1条（著作権）

株式会社エフイーシー（以下、当社といいます。）および／あるいはその許諾者が著作権を有する本ソフトウェアに関し、当社はお客様へライセンスするために必要な権利を有しています。お客様による本ソフトウェアの使用に関して、お客様にはこの契約中で許諾される以外は何らの権利も発生せず、本ソフトウェアに関する全ての権利は当社および／あるいはその許諾者に帰属します。

第2条（ライセンス）

本ソフトウェアは、弊社が許諾したライセンス数です。

第3条（使用許諾）

本ソフトウェアは、本製品のソフトウェア（バージョンアップ版も含まれます）であり、お客様は、この契約の内容に従うことを条件に、本製品でのみ本ソフトウェアを使用することができます。

第4条（使用の制限）

1. お客様は、本ソフトウェア及び付属するドキュメントを複製することはできません。
2. お客様は、本ソフトウェアの変更または改造を行うことはできません。
3. お客様は、本ソフトウェアについて、リバースエンジニアリング、逆コンパイル、逆アセンブル、解析を行うことはできません。
4. お客様は、本ソフトウェアを、有償・無償を問わず第三者に貸与することはできません。

第5条（保証の範囲）

1. お客様は、本ソフトウェアがいかなる状況においても不具合を生じることなく動作継続できるものではないことをあらかじめご了承ください。
2. 本ソフトウェアの不具合により本製品中に保存されたデータが変化・消失した場合でも、保証致しません。
3. 当社は、お客様が本ソフトウェアを使用することあるいは使用できないことから生じる偶発的あるいは間接的な損害、または受けられるべき救済の損失、得べかりし利益の損失、その他使用に起因して生じるいかなる損害に対しても責任を負いません。

4. お客様の変更または改造により、本ソフトウェアに何らかの欠陥が生じたとしても、当社は保証致しません。また、変更または改造の結果、万一お客様に損害を生じたとしても、当社は責任を負いません。
5. この契約に基づく当社の責任は、本製品のご購入に際しお客様が実際に支払われた金額を上限とします。

第6条（譲渡）

お客様は、この契約書と共に本ソフトウェア、付属するドキュメント並びに本製品の全てを譲渡し、譲渡を受ける方がこの契約の条件に同意した場合に限り、この契約に基づくお客様の権利を譲渡することができます。

第7条（契約の終了）

1. お客様は、いかなる時でも本ソフトウェア並びに付属するドキュメントを廃棄することによって、この契約を終了させることができます。
2. お客様がこの契約の内容に違反した場合、当社はこの契約を解除し、お客様における本ソフトウェアの使用を終了させることができます。その場合、お客様は本ソフトウェア（すべてのコピーを含みます）並びに付属するドキュメントを廃棄しなければなりません。

第8条（その他）

本ソフトウェア及び本製品を海外に持ち出す場合には、お客様は日本国外国為替及び外国貿易法、米国輸出管理法およびその他の国の法令を遵守しなければなりません。またこの契約は、日本法により解釈され、統治されるものとします。

第8条（第三者受益者）

本ソフトウェアには、当社以外の受諾者が権利を有するソフトウェアが含まれています。お客様が本契約に違反した場合、当社に加え、当該許諾者が直接お客様へ損害賠償を請求する権利があります。

1-1.他社製品の登録商標および商標についてのお知らせ

- Microsoft Windows、Windows10 IoT、Visual Studio、Windows Azure は、米国 Microsoft Corporation の米国およびその他の国における登録商標です。
- I2C は、Royal Philips、NXP のオランダ及びその他の国における登録商標です。
- Raspberry Pi は、英国ラズベリーパイ財団及びその他の国における登録商標です。
- DragonBoard は、米国 Qualcomm 及び開発関係会社及びの他の国における登録商標です。
- その他の名称及び商品名称は、各社の商標または登録商標です。

1-2.参考資料について

本ソフトウェアは、各メーカーの以下の資料をベースに構築されています。

- ・ I2Cバスインターフェース UM10204
http://www.nxp.com/documents/user_manual/UM10204_JA.pdf
- ・ 12bitA/D コンバータ ADC121C021
<http://www.ti.com/lit/ds/symlink/ad121c021.pdf>
- ・ デジタル光センサ TSL2561
<http://ams.com/eng/Products/Light-Sens...nsors/TSL2560/TSL2560-TSL2561-Datasheet>
- ・ 3軸ジャイロセンサ ITG 3200
<http://www.sparkfun.com/datasheets/Sensors/Gyro/PS-ITG-3200-00-01.4.pdf>
- ・ タッチセンサー MPR121
<https://www.sparkfun.com/datasheets/Components/MPR121.pdf>
- ・ 12bitD/A コンバーター MCP4725
<http://ww1.microchip.com/downloads/en/DeviceDoc/22039d.pdf>
- ・ 気圧・温度・湿度センサ BME280
https://ae-bst.resource.bosch.com/media/_tech/media/datasheets/BST-BME280_DS001-11.pdf
- ・ ステッピングモーターコントローラ PCA9629APW
http://www.nxp.com/documents/data_sheet/PCA9629A.pdf
- ・ ジェスチャーセンサ PAJ7620U2
http://www.pixart.com/upload/PAJ7620U2_GDS_v1.0_29032016_20160623194552.pdf
- ・ 液晶モジュールコントローラ
<https://www.seeedstudio.com/Grove-LCD-RGB-Backlight-p-1643.html>
- ・ 8×8マトリクス LED コントローラ PCA9622
http://www.nxp.com/documents/data_sheet/PCA9622.pdf?pspll=1
- ・ 3軸加速度センサ ADXL345
<http://www.analog.com/media/en/technical-documentation/datasheets/ADXL345.pdf>

1-3.I2C デバイス接続上の注意点

1.ロジック電圧の注意点

I2C デバイスは、3.3V で動作するデバイスと 5V で動作するデバイスの 2 種類存在します。3.3V のデバイスを 5V で動作するデバイスに接続しても動作しません。また、5V を印加したデバイスを 3.3V 動作のデバイスに接続すると 3.3V 動作デバイスが壊れます。

これを回避する為には、3.3V のデバイスと 5V デバイスを接続する為のロジックレベルコンバーターが必要と成ります。

TXB0108 等のレベルコンバーターや BSS138 等の FET を用いたレベルコンバーターが数多く存在します。適切なレベルコンバーターを用いて正しく接続して下さい。

(参考)

・TXB0108 8-Bit Bidirectional Voltage-Level Translator

<http://www.tij.co.jp/jp/lit/ds/symlink/txb0108.pdf>

2.複数デバイス接続時の注意点

I2C デバイスを複数個接続する場合、3.3V 又は 5V の電源共有能力に注意して下さい。Raspberry Pi 等の 3.3V 電源を使用する場合、Raspberry Pi 内部で 5V から 3.3V への変換回路を用いており、基板内のデバイスを動かす程度の容量しか有りません。その為、この 3.3V を I2C デバイスに直接供給すると、3.3V の電源が供給不足となり動作不安定の原因と成ります。

3.I2C デバイスの接続距離の注意点

I2C インターフェースは、そもそも同一基板内の近距離に配置されたデバイスを制御する為のインターフェースで、I2C ラインの負荷容量も合計で 400pF 以内の制限が有ります。数 m の距離を延長しての接続は、接続する線の線間容量が増大し規格値を大幅に超えます。また、ノイズ等の対策も必要と成ります。

数 m の距離を延長する場合には、I2C ラインに I2C バスドライバー等を挿入し、ノイズ対策を完璧に実施して下さい。

(参考) PCA9600 (Dual bidirectional bus buffer)

http://www.nxp.com/documents/data_sheet/PCA9600.pdf

2.I2C とは・・・

I2C インターフェースは、1980 年代にフィリップスが提唱した、基板内の各種チップを接続する為のシンプルなバスシステムで、*Inter-Integrated Circuit* の略です。

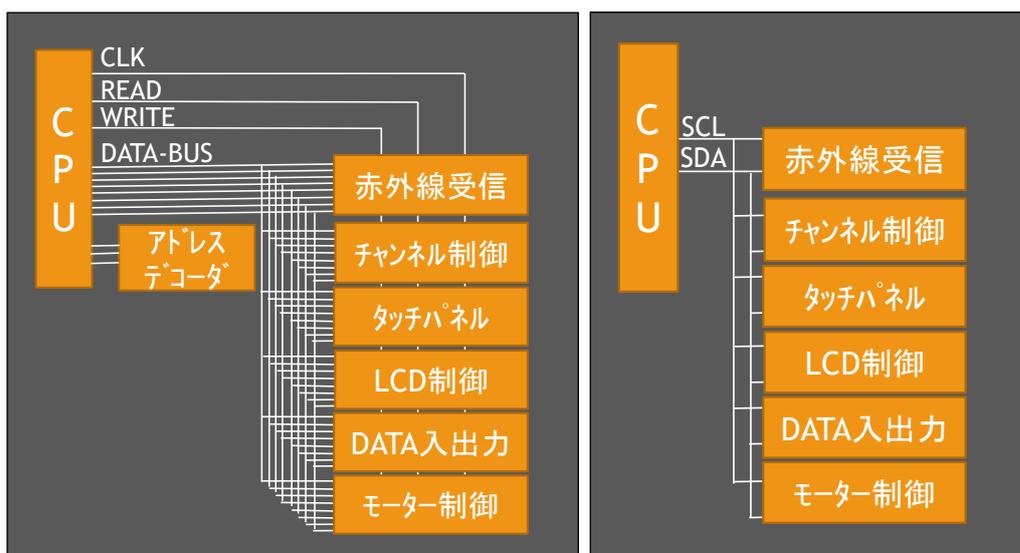
従来のバスシステムでは、各チップやデバイスを接続する為には、データのやり取りをする為のデータバスを 8bit の場合 8 本、16bit の場合 16 本が必要で、その以外に、チップやデバイスを指定する CS (チップセレクト)、読み書きを指定する RD (リード)、WR (ライト)、及び、そのタイミングを合わせる為に CLK (クロック) 信号と CPU とチップやデバイス間の情報伝達に数多くの配線が必要としました。

この従来のバスの場合、パラレルでデータの受け渡しを行う為、CPU 性能に匹敵した高速でのデータの受け渡しが行える反面、プリント基板内に数多くの配線を張り巡らす必要があり、基板内の実装面積の大半がこれらの配線で占められていました。

この課題に対し、I2C インターフェースは、CPU 性能まで必要としないチップやデバイスをシリアル通信で接続する事で、SCL (クロック)、SDA (データ) の 2 本で、通信を可能とした規格です。

さらに、I2C は、マルチドロップ (芋づる式) での接続が可能で最大 112 個のデバイスを接続可能です。

配線数が大幅に減少する為、小型化、軽量化が可能となり数多くの家電製品、携帯電話、タブレット、ノートパソコン等に広く使用されており、現在 50 社以上から 1000 種類を超える、I2C インターフェース仕様のチップ、デバイスが販売されています。



従来の接続方法（家庭用 DVD デッキ例）

I2C での接続方法

2-1.I2C センサードライバーとは・・・

I2C インターフェースでの各チップ、デバイスの利用は家電製品や携帯電話等の組み込みシステムに数多く採用されています。

I2C インターフェースに対応しているデバイスは、 μ コントローラが内臓され、I2C 通信や、デバイス自身の制御を行っており、I2C 通信で制御指示を受けると自ら制御やデータ収集等を行う事が出来ます。

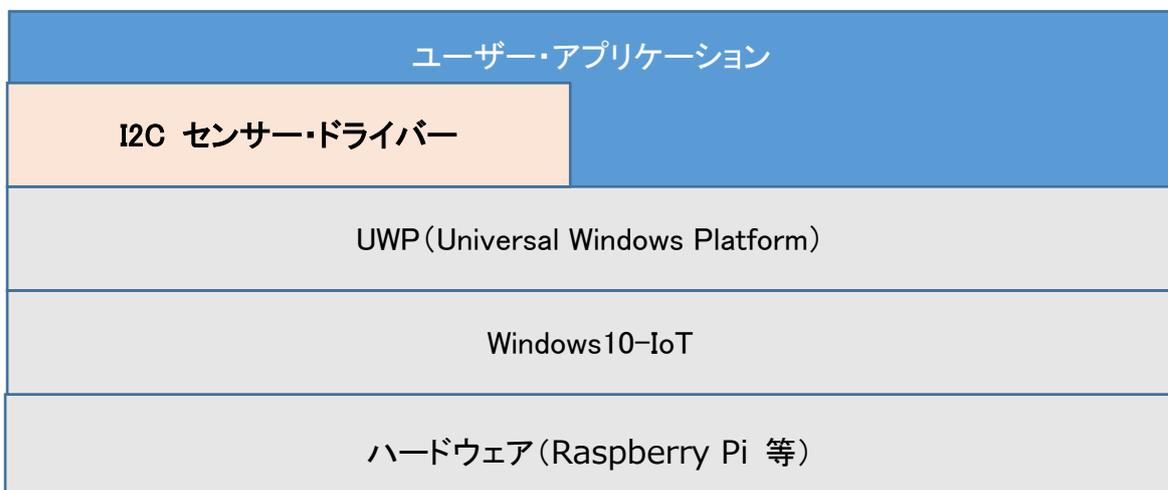
制御やデータ収集を自ら実施する為には、その動作条件をデバイスに準備されたレジスタにマスターCPU が設定しなければならず、又、データ収集もデータが格納されたレジスタよりマスターCPU が読出しを行う必要が有ります。

しかし、各デバイスに準備されたレジスタは、レジスタ数を極力減らし、又 I2C 通信回数を減らす為、ひとつのレジスタを複数の設定機能に分割し、レジスタの使用効率を上げす為、bit 単位での設定が必要となり、非常に複雑な設定となります。

この設定を Windows10-IoT 上の言語、Visual Studio2015 の VB 等の高級言語では、Bit 処理が面倒で、Bit 処理の知識が必須となります。

これらの知識を必要とせず、VB 等の広く開発で利用されている言語をもちいて開発可能とした API(アプリケーション・インターフェース)が I2C センサー・ドライバーです。

(I2C センサー ドライバーの位置づけ)



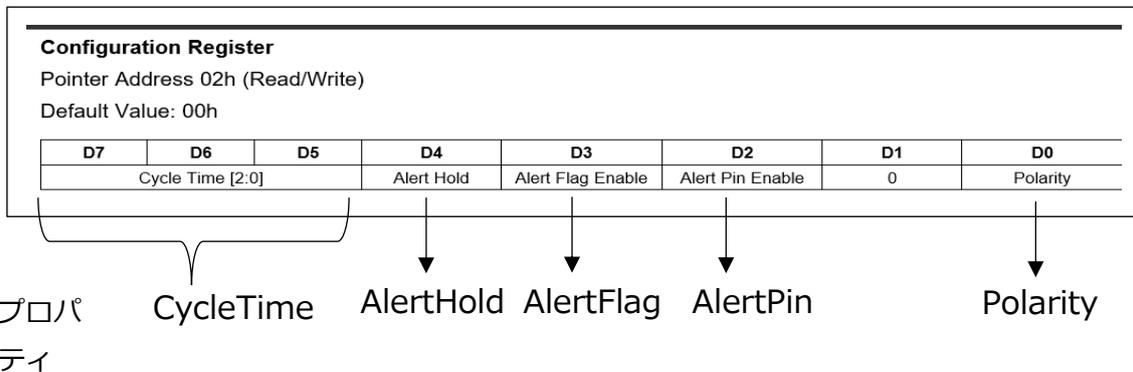
2-2.I2C センサードライバーの使い方

例) ADC121C021 の Configuration Register (Pointer Address 02h) の場合

このレジスタは、アナログデータをデジタルデータに変換する間隔 (Conversion interval) 。アラート設定値の範囲を超えたデータの入力が有った場合、そのアラートを保持して置くか (Alert Hold) ,アラートフラグを許可するか (Alert Flag Enable) ,アラートピン出力を許可するか? (Alert Pin Enable) とアラート発生時のピン出力は、アラート発生時には High か Low 出力かの 5 種類の設定を 1Byte で構成されています。その為、これらの条件を設定する場合、各 Bit の ON/OFF の組み合わせで、8bit のデータを構成して ADC121C021 の Pointer Address 02h へ Write データとして設定しなければなりません。

この紛らわしい、Bit 処理を専用のプロパティを個別に準備した I2C センサー・ドライバーを用いる事により、bit の組み合わせや Pointer Address を気にする事無く、開発が可能です。

機能毎にプロパティは準備されています。例えば、プロパティの AlertFlag に 1 を設定し write すれば、他の設定は、そのままに Pointer Address 02 の D3 の bit のみを 1 にする事ができます。



※ 設定する値については、各デバイスの Data Sheet をご覧ください。

この様に I2C デバイスを使う上で、複数の bit 構成での各種設定や、設定されるデータ等を I2C インターフェース・ドライバーにて、個別のプロパティで扱う事で、bit 単位での処理を全く意識する事無く、プログラム開発が可能と成ります。

また、プロパティ名が Data Sheet の説明と同じ用語を用いる事で、開発の容易度が増します。

2-3.I2C センサー・ドライバーの使用上の注意

I2C センサー・ドライバーは、各種設定の組み合わせが 1Byte 内の凝縮されており、これらを、1bit 単位ではなく、プロパティ単位で設定、読み出しを可能としています。これらのプロパティに設定する内容は、Data Sheet に記載された内容を目的に沿った値を設定、もしくは読み出しを行って下さい。

特に、データの読み出し時の単位変換は、汎用性を高める為に行っておりません。I2C でデバイスのレジスタに書きこまれている内容そのものを読み出します。

例) 温度・湿度・気圧センサー BME280 の場合

| Register Name | Address | bit7 | bit6 | bit5 | bit4 | bit3 | bit2 | bit1 | bit0 | Reset state | |
|------------------|-------------|------------------|------|------|-------------|------|--------------|-------------|------|-------------|------|
| hum_lsb | 0xFE | hum_lsb<7:0> | | | | | | | | 0x00 | |
| hum_msb | 0xFD | hum_msb<7:0> | | | | | | | | 0x80 | |
| temp_xlsb | 0xFC | temp_xlsb<7:4> | | | | 0 | 0 | 0 | 0 | 0x00 | |
| temp_lsb | 0xFB | temp_lsb<7:0> | | | | | | | | 0x00 | |
| temp_msb | 0xFA | temp_msb<7:0> | | | | | | | | 0x80 | |
| press_xlsb | 0xF9 | press_xlsb<7:4> | | | | 0 | 0 | 0 | 0 | 0x00 | |
| press_lsb | 0xF8 | press_lsb<7:0> | | | | | | | | 0x00 | |
| press_msb | 0xF7 | press_msb<7:0> | | | | | | | | 0x80 | |
| config | 0xF5 | t_sb[2:0] | | | filter[2:0] | | | spi3w_en[0] | | 0x00 | |
| ctrl_meas | 0xF4 | osrs_t[2:0] | | | osrs_p[2:0] | | | mode[1:0] | | 0x00 | |
| status | 0xF3 | measuring[0] | | | | | im_update[0] | | | | 0x00 |
| ctrl_hum | 0xF2 | osrs_h[2:0] | | | | | | | | 0x00 | |
| calib26..calib41 | 0xE1...0xF0 | calibration data | | | | | | | | individual | |
| reset | 0xE0 | reset[7:0] | | | | | | | | 0x00 | |
| id | 0xD0 | chip_id[7:0] | | | | | | | | 0x60 | |
| calib00..calib25 | 0x88...0xA1 | calibration data | | | | | | | | individual | |

| | | | | | | | |
|------------|-------------------------------------|-------------------------------|-----------------------------------|-----------------------------|-------------------------------|----------------------|---------------------|
| Registers: | Reserved registers do not change | Calibration data read only | Control registers read / write | Data registers read only | Status registers read only | Chip ID read only | Reset write only |
| Type: | | | | | | | |

BME280 のレジスタ構成は上図の通りです。ここで各種初期設定が完了しデータを収集した場合、温度データは、温度の情報はレジスタのアドレス 0xFA 0xFB 0xFc の3つのレジスタに分割して格納されます。I2C センサー・ドライバーでは、この3つのレジスタよりデータを読み出し、ビットシフト処理を行い、0xE1~0xF0 の温度補正データを読み出し、補正演算を行った結果を ReadTemperature メソッドに返します。

この返したデータは補正処理が完了した 16 進数表記のままです。ユーザーがこの 16 進数の得られたデータから、摂氏温度や華氏温度等への単位変換等、利用するアプリケーションに合わせて変換を行って下さい。

I2C センサー・ドライバーでは、利用するアプリケーションの広さに対応する為、生データを得る迄の機能とし、その後の単位変換等はユーザーが実施可能とする事で応用範囲を広げています。

2-4.開発環境

開発を行う為には、以下の環境の準備が必要となります。

開発用 PC では Windows10 と Visual Studio 2015 の環境が必要です。

1. Hard Ware

1-1.Raspberry Pi3 Model-B

1-2.HDMI 入力端子付きのディスプレイと HDMI (Type-A) ケーブル

1-3.有線 LAN ケーブル (設定により無線 LAN も可)

1-4.microSD メモリーカード (Class 10/32GB 以上を推奨)

1-5.USB (ないし Bluetooth) キーボード

1-6.USB (ないし Bluetooth) マウス

1-7.給電用の USB 電源アダプター (5V/2.5A) と USB Micro-B ケーブル

2. I2C デバイス

接続を行う、I2C 仕様準拠のデバイス

3. O/S

マイクロソフトの Windows10IoT のサイトより Raspberry Pi3 用の O/S をダウンロードし、ISO イメージファイルを microSD カードに書き込み、SD カードを Raspberry Pi3 に差し込み起動します。

<https://developer.microsoft.com/en-us/windows/IoT/GetStarted>

詳細な手順についてはマイクロソフトの Windows10-IoT サイト及び、関連記事、書籍等を参考に行ってください

2-5.本資料の使い方

本資料は、「I2C インターフェース・ドライバー」で提供される機能の説明です。

1.プロパティ

I2C デバイスの内部レジスタを、機能毎にプロパティに分割しています。まず、該当デバイスの Data Sheet を良くお読みになり、必要とする機能を実現する為には、どのレジスタに何を設定すれば良いか？の設定条件を決定して下さい。

その決定した設定条件をプロパティに設定して下さい。レジスタと該当する bit への処理は「I2C インターフェース・ドライバー」がプロパティ名より自動判断し、READ/WRITE を行います。

Read メソッドを実行後には、プロパティに I2C デバイスから得た値が格納されています。

2.メソッド

プロパティに設定された内容を実際に I2C デバイスへの READ/WRITE を行う関数です。

2-1.「Init」メソッド

I2C デバイスと通信の確立を行うメソッドで、通信開始する場合必ず一度実行して下さい。

2-2.「Read」メソッド

I2C デバイスのレジスタよりデータを読み出すメソッドです。プロパティに読み出した結果がセットされます。

2-3.「Write」の3種類準備されております

I2C デバイスのレジスタにデータを書き込みメソッドです。初期設定や、出力条件変更等は、このメソッドを使用して下さい。

3.定数

該当する I2C デバイスのレジスタ番号を定義しています。

-

3. 提供する I2C センサー・ドライバーの種類

I2C センサー・ドライバーで提供する I2C デバイス

汎用性を鑑み、以下の I2C インターフェース対応デバイスのドライバーソフトを提供します。

| | デバイスの種類 | デバイス名 | 仕様 | 応用例 |
|----|--------------------------|------------|-----------------|---------------------------------------|
| 1 | 12bitA/D コンバータ | ADC121C021 | 12bit/1ch | アナログ電圧計測 |
| 2 | デジタル光センサ | TSL2561 | 0.1~40,000LUX | 自然光,IR 計測.照明の ON/OFF |
| 3 | 3 軸ジャイロセンサ | ITG 3200 | XYZ、±2000°/S | 物・ロボットの傾きの移動量(振動) |
| 4 | タッチセンサー | MPR121 | 静電容量検出 8ch | 操作パネルの入力 |
| 5 | 12bitD/A コンバーター | MCP4725 | 12bit/ch | アナログ制御電圧の出力 |
| 6 | 気圧・温度・湿度 センサ | BME280 | 温度、湿度、気圧 計測 | 気温-40~80℃、 300~1100hPa,0~100%、環境測定 |
| 7 | ステッピングモーター コントローラ | PCA9629APW | 4 phase stepper | モーター駆動、位置決め |
| 8 | ジェスチャーセンサー | PAJ7620U2 | 9 つの動作 | スイッチを直接操作できない環境 での、簡易動作指示等 |
| 9 | 液晶モジュールコントローラ | PIC16F77A | 16x2 LCD | 表示装置 |
| 10 | 16chPWM コントローラ | PCA9685 | 16ch | 輝度調整、ラジコンサーボ制御 |
| 11 | 8x8 マトリックス LED コントローラ | PCA9622 | 8x8 マトリックス | 簡易表示装置、操作指示 方向指示等 |
| 12 | 3 軸加速度センサ | ADXL345 | ±16g(重力,衝撃) | 傾き量、衝突方向・強度・転倒検出 |

3-1.12bit A/D コンバータ ADC121C021

変更履歴

2016年9月 12日 第一版 新規作成

1. 概要

12ビットアナログ - デジタルアラート機能付きコンバータ(ADC121C021)と I2C 接続するための機能を提供します。

2. 機能一覧

プロパティ

| 機能名 | 機能説明 |
|-----------------|--|
| Polarity | このビットは、ALERT 出力ピンのアクティブレベルの極性を設定します。 0: アクティブ・ローの ALERT ピンを設定します。 1: アクティブ・ハイに ALERT ピンを設定します。 |
| AlertPinEnable | 0: ALERT 出力ピンを無効にします。ピンが無効になっているときに ALERT 出力はハイインピーダンスになります。 1: ALERT 出力ピンを有効にします。 |
| AlertFlagEnable | 0: 変換結果レジスタにおける障害のアラートステータスビット[D15]。 1: 変換結果レジスタ内のアラートステータスビット[D15]を有効にします。 |
| AlertHold | 0: アラートは、意志、自己明確なとき、ヒステリシス・レジスタの値以上により制限内で測定された電圧が移動します。 1: アラートが自己明確ではないだろうと 1 がアラートステータス・レジスタ内のアラート高いフラグまたは警告低フラグに書き込まれたときにのみクリアされます。 |

| 機能名 | 機能説明 |
|-------------------|---|
| CycleTime | 0: 自動モード無効 1: Tconvert x 32 2: Tconvert x 64 3: Tconvert x 128 4: Tconvert x 256 5: Tconvert x 512 6: Tconvert x 1024 7: Tconvert x 2048 |
| ResultValue | 最新の変換結果 |
| AlertFlag | 最新の変換結果のアラートフラグ |
| OverRangeAlert | VHIGH リミットレジスタに格納された VHIGH の限界を超えた場合、ビットは 1 に設定されています。 |
| UnderRangeAlert | VLOW リミットレジスタに格納され VLOW 限界を下回ったときにビットが 1 に設定されています。 |
| VLOWLimit | アラート条件を決定するために使用される下限閾値 |
| VHIGHLimit | アラート条件を決定するために使用される上限閾値 |
| Hysteresis | アラート条件を決定するために使用されるヒステリシス値 |
| LowestConversion | 最低の変換結果 |
| HighestConversion | 最高の変換結果 |
| ReadBuff | 読込生データ |

メソッド

| 機能名 | 機能説明 |
|-------|---------------------------------------|
| Init | I2C バスコントローラに指定された接続設定を使って非同期的に接続します。 |
| Read | 指定レジスタ読込 |
| Write | 指定レジスタ書込 |

定数

| 機能名 | 機能説明 | 定数値 |
|-----------------|--------------------|------|
| REG_ADDR.RESULT | Conversion Result | 0x00 |
| REG_ADDR.ALERT | Alert Status | 0x01 |
| REG_ADDR.CONFIG | Configuration | 0x02 |
| REG_ADDR.LIMITL | Low Limit | 0x03 |
| REG_ADDR.LIMITH | High Limit | 0x04 |
| REG_ADDR.HYST | Hysteresis | 0x05 |
| REG_ADDR.CONVL | Lowest Conversion | 0x06 |
| REG_ADDR.CONVH | Highest Conversion | 0x07 |

Polarity プロパティ

概要

Configuration レジスタの Polarity の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.CONFIG` を指定した時に保持されます。

Write メソッドで `REG_ADDR.CONFIG` を指定した時に Polarity 値を使用して書込まず。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.CONFIG` を指定した時に保持した値
または、SET 処理をした値

AlertPinEnable プロパティ

概要

Configuration レジスタの AlertPinEnable の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.CONFIG` を指定した時に保持されます。

Write メソッドで `REG_ADDR.CONFIG` を指定した時に AlertPinEnable 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.CONFIG` を指定した時に保持した値
または、SET 処理をした値

AlertFlagEnable プロパティ

概要

Configuration レジスタの AlertFlagEnable の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.CONFIG` を指定した時に保持されます。

Write メソッドで `REG_ADDR.CONFIG` を指定した時に AlertFlagEnable 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.CONFIG` を指定した時に保持した値
または、SET 処理をした値

AlertHold プロパティ

概要

Configuration レジスタの AlertHold の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.CONFIG` を指定した時に保持されます。

Write メソッドで `REG_ADDR.CONFIG` を指定した時に AlertHold 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.CONFIG` を指定した時に保持した値
または、SET 処理をした値

CycleTime プロパティ

概要

Configuration レジスタの CycleTime の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.CONFIG` を指定した時に保持されます。

Write メソッドで `REG_ADDR.CONFIG` を指定した時に CycleTime 値を使用して書込ま
す。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.CONFIG` を指定した時に保持した値
または、SET 処理をした値

ResultValue プロパティ

概要

Conversion Result レジスタの Conversion Result の値を取得します。

処理内容

Read メソッドで `REG_ADDR.RESULT` を指定した時に保持されます。

SET 処理

なし

GET 処理

Read メソッドで `REG_ADDR.RESULT` を指定した時に保持した値

AlertFlag プロパティ

概要

Conversion Result レジスタの AlertFlag の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.RESULT` を指定した時に保持されます。

SET 処理

なし

GET 処理

Read メソッドで `REG_ADDR.RESULT` を指定した時に保持した値

OverRangeAlert プロパティ

概要

Alert Status レジスタの Over Range Alert の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.ALERT` を指定した時に保持されます。

Write メソッドで `REG_ADDR.ALERT` を指定した時に OverRangeAlert 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.ALERT` を指定した時に保持した値
または、SET 処理をした値

UnderRangeAlert プロパティ

概要

Alert Status レジスタの Under Range Alert の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.ALERT` を指定した時に保持されます。

Write メソッドで `REG_ADDR.ALERT` を指定した時に UnderRangeAlert 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.ALERT` を指定した時に保持した値
または、SET 処理をした値

VLOWLimit プロパティ

概要

VLOW Alert Limit レジスタの VLOWLimit の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.LIMITL` を指定した時に保持されます。

Write メソッドで `REG_ADDR.LIMITL` を指定した時に UnderRangeAlert 値を使用して書込まれます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.LIMITL` を指定した時に保持した値
または、SET 処理をした値

VHIGHLimit プロパティ

概要

VHIGH Alert Limit レジスタの VHIGHLimit の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.LIMITH` を指定した時に保持されます。

Write メソッドで `REG_ADDR.LIMITH` を指定した時に VHIGHLimit 値を使用して書込ま
す。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.LIMITH` を指定した時に保持した値
または、SET 処理をした値

Hysteresis プロパティ

概要

VHYST Alert Hysteresis レジスタの Hysteresis の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.HYST` を指定した時に保持されます。

Write メソッドで `REG_ADDR.HYST` を指定した時に Hysteresis 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.HYST` を指定した時に保持した値
または、SET 処理をした値

LowestConversion プロパティ

概要

VMIN Lowest Conversion レジスタの LowestConversion の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.CONVL` を指定した時に保持されます。

Write メソッドで `REG_ADDR.CONVL` を指定した時に LowestConversion 値を使用して書込まれます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.CONVL` を指定した時に保持した値
または、SET 処理をした値

HighestConversion プロパティ

概要

VMAX Highest Conversion レジスタの HighestConversion の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.CONVH` を指定した時に保持されます。

Write メソッドで `REG_ADDR.CONVH` を指定した時に HighestConversion 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.CONVH` を指定した時に保持した値
または、SET 処理をした値

ReadBuff プロパティ

概要

Read メソッド時の読込データをセットします。

処理内容

Read メソッド時に保持されます。

SET 処理

なし

GET 処理

Read メソッド時に保持した値

Init メソッド

概要

I2C バスコントローラに指定された接続設定を使って非同期的に接続します。

引数

| 引数名 | 型 | 説明 |
|---------|------|-----------|
| Address | byte | I2C のアドレス |

戻り値

なし

Read メソッド

概要

最新の変換結果を読込みます。

引数

| 引数名 | 型 | 説明 |
|----------|------|----|
| REG_ADDR | byte | |

戻り値

Status と同じ

1: 正常

1 以外: 異常

読込処理

| REG_ADDR 値 | 読込プロパティ |
|------------|-------------------|
| RESULT | AlertFlag |
| | ResultValue |
| ALERT | OverRangeAlert |
| | UnderRangeAlert |
| CONFIG | CycleTime |
| | AlertHold |
| | AlertFlagEnable |
| | AlertPinEnable |
| | Polarity |
| LIMITL | VLOWLimit |
| LIMITH | VHIGHLimit |
| HYST | Hysteresis |
| CONVL | LowestConversion |
| CONVH | HighestConversion |

Write メソッド

概要

引数 RegSelect に関するプロパティを編集し書き込みます。

引数

| 引数名 | 型 | 説明 |
|----------|------|----|
| REG_ADDR | byte | |

戻り値

Status

1: 正常

1 以外: 異常

書込処理

| REG_ADDR 値 | 読込プロパティ |
|------------|-------------------|
| ALERT | OverRangeAlert |
| | UnderRangeAlert |
| CONFIG | CycleTime |
| | AlertHold |
| | AlertFlagEnable |
| | AlertPinEnable |
| | Polarity |
| LIMITL | VLOWLimit |
| LIMITH | VHIGHLimit |
| HYST | Hysteresis |
| CONVL | LowestConversion |
| CONVH | HighestConversion |

サンプルプログラム

Imports I2CLibrary

Public NotInheritable Class ADC121C021Page

Inherits Page

Private Const ADC121C021_Address As Byte = &H50

Private ADC121C021 As ADC121C021 = New ADC121C021()

Private WithEvents RefreshTimer As DispatcherTimer = New DispatcherTimer()

Public Sub New()

InitializeComponent()

Call Init()

End Sub

Private Async Sub Init()

Await ADC121C021.Init(ADC121C021_Address)

' HighestConversion初期化

ADC121C021.HighestConversion = 0

ADC121C021.Write(ADC121C021.REG_ADDR.CONVH)

' LowestConversion初期化

ADC121C021.LowestConversion = &HFFF

ADC121C021.Write(ADC121C021.REG_ADDR.CONVL)

' コンフィグ書込

ADC121C021.AlertFlagEnable = True

ADC121C021.AlertPinEnable = True

ADC121C021.Write(ADC121C021.REG_ADDR.CONFIG)

' 上限値

ADC121C021.VHIGHLimit = 2000

ADC121C021.Write(ADC121C021.REG_ADDR.LIMITH)

' 下限値

ADC121C021.VLOWLimit = 500

ADC121C021.Write(ADC121C021.REG_ADDR.LIMITL)

End Sub

Private Sub btnStart_Click(sender As Object, e As RoutedEventArgs)

RefreshTimer.Interval = TimeSpan.FromMilliseconds(100)

RefreshTimer.Start()

End Sub

Private Sub Measurement(sender As Object, e As Object) Handles RefreshTimer.Tick

RefreshTimer.Stop()

ADC121C021.Read(ADC121C021.REG_ADDR.RESULT)

```

Debug.WriteLine("{0:F} {1}", ADC121C021.ResultValue, ADC121C021.AlertFlag)

ADC121C021.Read(ADC121C021.REG_ADDR.ALERT)
Debug.WriteLine("{0} {1}", ADC121C021.OverRangeAlert, ADC121C021.UnderRangeAlert)

ADC121C021.Read(ADC121C021.REG_ADDR.CONVH)
Debug.WriteLine("{0:F}", ADC121C021.HighestConversion)

ADC121C021.Read(ADC121C021.REG_ADDR.CONVL)
Debug.WriteLine("{0:F}", ADC121C021.LowestConversion)

RefreshTimer.Start()
End Sub

Private Sub btnStop_Click(sender As Object, e As RoutedEventArgs)
    RefreshTimer.Stop()

    ADC121C021.HighestConversion = 0
    ADC121C021.Write(ADC121C021.REG_ADDR.CONVH)

    ADC121C021.LowestConversion = &HFFF
    ADC121C021.Write(ADC121C021.REG_ADDR.CONVL)
End Sub

Private Sub btnBack_Click(sender As Object, e As RoutedEventArgs)
    RefreshTimer.Stop()
    ADC121C021.close()
    App.rootFrame.Navigate(GetType(MenuPage))
End Sub

End Class

```

3-2.デジタル光センサー TSL2561

変更履歴

2016年10月12日 第一版 新規作成

1. 概要

デジタル光センサー(TSL2561)とI2C 接続するための機能を提供します。

2. 機能一覧

プロパティ

| 機能名 | 機能説明 |
|------------|-------------------------------|
| POWER | パワーアップ／パワーダウンを指定します。 |
| GAIN | 低ゲインモードと高ゲインモードの間でゲインを切り替えます。 |
| Manual | 積分サイクルの開始／停止を指定します。 |
| INTEG | 積分時間を指定します。 |
| THRESHLOW | ADC チャンネル 0 低しきい値を取得または設定します。 |
| THRESHHIGH | ADC チャンネル 0 高しきい値を取得または設定します。 |
| INTR | INTR コントロールを取得または設定します。 |
| PERSIST | 割り込みの持続性を取得または設定します。 |
| PARTNO | 品番識別情報を取得します。 |
| REVNO | リビジョン番号を取得します。 |
| DATA0 | ADC チャンネル 0 のデータを取得します。 |
| DATA1 | ADC チャンネル 1 のデータを取得します。 |
| ReadBuff | 読込生データ |

メソッド

| 機能名 | 機能説明 |
|-------|---------------------------------------|
| Init | I2C バスコントローラに指定された接続設定を使って非同期的に接続します。 |
| Read | 指定レジスタ読込 |
| Write | 指定レジスタ書込 |

定数

| 機能名 | 機能説明 | 定数値 |
|---------------------|-----------------|------|
| REG_ADDR.CONTROL | 基本的な機能の制御 | 0x00 |
| REG_ADDR.TIMING | 積分時間／ゲイン・コントロール | 0x01 |
| REG_ADDR.THRESHLOW | 低割り込みしきい値 | 0x02 |
| REG_ADDR.THRESHHIGH | 高割り込みしきい値 | 0x04 |
| REG_ADDR.INTERRUPT | 割り込み制御 | 0x06 |
| REG_ADDR.CRC | 工場テスト用 | 0x08 |
| REG_ADDR.ID | 部品番号／ Rev ID | 0x0A |
| REG_ADDR.DATA0 | ADC チャンネル 0 データ | 0x0C |
| REG_ADDR.DATA1 | ADC チャンネル 1 データ | 0x0E |

| 機能名 | 機能説明 | 定数値 |
|------------------------|---------|------|
| Integration_Time.ms013 | 13.7 ms | 0x00 |
| Integration_Time.ms101 | 101 ms | 0x01 |
| Integration_Time.ms402 | 402 ms | 0x02 |
| Integration_Time.NA | N/A | 0x03 |

| 機能名 | 機能説明 | 定数値 |
|---|--|------|
| Interrupt_Control.Interrupt_output_disabled | Interrupt output disabled | 0x00 |
| Interrupt_Control.Level_Interrupt | Level Interrupt | 0x01 |
| Interrupt_Control.SMBAlert_compliant | SMBAlert compliant | 0x02 |
| Interrupt_Control.Test_Mode | Test Mode: Sets interrupt and functions as mode 10 | 0x03 |

POWER プロパティ

概要

Control レジスタの POWER の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.CONTROL` を指定した時に保持されます。

Write メソッドで `REG_ADDR.CONTROL` を指定した時に POWER 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.CONTROL` を指定した時に保持した値
または、SET 処理をした値

GAIN プロパティ

概要

TIMING レジスタの GAIN の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR`. TIMING を指定した時に保持されます。

Write メソッドで `REG_ADDR`. TIMING を指定した時に GAIN 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR`. TIMING を指定した時に保持した値
または、SET 処理をした値

Manual プロパティ

概要

TIMING レジスタの Manual の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR`. TIMING を指定した時に保持されます。

Write メソッドで `REG_ADDR`. TIMING を指定した時に Manual 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR`. TIMING を指定した時に保持した値
または、SET 処理をした値

INTEG プロパティ

概要

TIMING レジスタの INTEG の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR`. TIMING を指定した時に保持されます。

Write メソッドで `REG_ADDR`. TIMING を指定した時に INTEG 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR`. TIMING を指定した時に保持した値
または、SET 処理をした値

THRESHLOW プロパティ

概要

THRESHLOWLOW、THRESHLOWHIGH レジスタの THRESHLOW の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.THRESHLOW` を指定した時に保持されます。

Write メソッドで `REG_ADDR.THRESHLOW` を指定した時に THRESHLOW 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.THRESHLOW` を指定した時に保持した値
または、SET 処理をした値

THRESHHIGH プロパティ

概要

THRESHHIGHLOW、THRESHHIGHHIGH レジスタの THRESHHIGH の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR`. THRESHHIGH を指定した時に保持されます。

Write メソッドで `REG_ADDR`. THRESHHIGH を指定した時に THRESHHIGH 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR`. THRESHHIGH を指定した時に保持した値
または、SET 処理をした値

INTR プロパティ

概要

INTERRUPT レジスタの INTR の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR. INTERRUPT` を指定した時に保持されます。

Write メソッドで `REG_ADDR. INTERRUPT` を指定した時に INTR 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR. INTERRUPT` を指定した時に保持した値
または、SET 処理をした値

PERSIST プロパティ

概要

INTERRUPT レジスタの PERSIST の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR_INTERRUPT` を指定した時に保持されます。

Write メソッドで `REG_ADDR_INTERRUPT` を指定した時に PERSIST 値を使用して書込ま
す。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR_INTERRUPT` を指定した時に保持した値
または、SET 処理をした値

PARTNO プロパティ

概要

ID レジスタの PARTNO の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR`. ID を指定した時に保持されます。

Write メソッドで `REG_ADDR`. ID を指定した時に PARTNO 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR`. ID を指定した時に保持した値
または、SET 処理をした値

REVNO プロパティ

概要

ID レジスタの REVNO の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR`. ID を指定した時に保持されます。

Write メソッドで `REG_ADDR`. ID を指定した時に REVNO 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR`. ID を指定した時に保持した値
または、SET 処理をした値

DATA0 プロパティ

概要

DATA0LOW、DATA0HIGH レジスタの DATA0 の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR`. DATA0 を指定した時に保持されます。

Write メソッドで `REG_ADDR`. DATA0 を指定した時に DATA0 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR`. DATA0 を指定した時に保持した値
または、SET 処理をした値

DATA1 プロパティ

概要

DATA1LOW、DATA1HIGH レジスタの DATA1 の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR`. DATA1 を指定した時に保持されます。

Write メソッドで `REG_ADDR`. DATA1 を指定した時に DATA1 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR`. DATA1 を指定した時に保持した値
または、SET 処理をした値

ReadBuff プロパティ

概要

Read メソッド時の読込データをセットします。

処理内容

Read メソッド時に保持されます。

SET 処理

なし

GET 処理

Read メソッド時に保持した値

Init メソッド

概要

I2C バスコントローラに指定された接続設定を使って非同期的に接続します。

引数

| 引数名 | 型 | 説明 |
|---------|------|-----------|
| Address | byte | I2C のアドレス |

戻り値

なし

Read メソッド

概要

最新の変換結果を読込みます。

引数

| 引数名 | 型 | 説明 |
|----------|------|----|
| REG_ADDR | byte | |

戻り値

Status と同じ

1: 正常

1 以外: 異常

読み処理

| REG_ADDR 値 | 読みプロパティ |
|------------|------------|
| INTERRUPT | INTR |
| | PERSIST |
| ID | PARTNO |
| | REVNO |
| CONTROL | POWER |
| TIMING | GAIN |
| | Manual |
| | INTEG |
| THRESHLOW | THRESHLOW |
| THRESHHIGH | THRESHHIGH |
| DATA0 | DATA0 |
| DATA1 | DATA1 |

Write メソッド

概要

引数 RegSelect に関するプロパティを編集し書き込みます。

引数

| 引数名 | 型 | 説明 |
|----------|------|----|
| REG_ADDR | byte | |

戻り値

Status

1: 正常

1 以外: 異常

書込処理

| REG_ADDR 値 | 書込プロパティ |
|------------|------------|
| INTERRUPT | INTR |
| | PERSIST |
| CONTROL | POWER |
| TIMING | GAIN |
| | Manual |
| | INTEG |
| THRESHLOW | THRESHLOW |
| THRESHHIGH | THRESHHIGH |

サンプルプログラム

```
Imports I2CLibrary
Public NotInheritable Class TSL2561Page
    Inherits Page

    Private Const TSL2561_Address As Byte = CType(TSL2561.SLAVE_ADDRESS.ADDR0, Byte)
    Private TSL2561 As TSL2561 = New TSL2561()
    Private WithEvents RefreshTimer As DispatcherTimer = New DispatcherTimer()

    Public Sub New()
        InitializeComponent()
        Call Init()
    End Sub

    Private Async Sub Init()
        Await TSL2561.Init(TSL2561_Address)

        TSL2561.INTR = TSL2561.Interrupt_Control.Interrupt_output_disabled
        TSL2561.PERSIST = &H0
        TSL2561.Write(TSL2561.REG_ADDR.INTERRUPT)

        TSL2561.GAIN = True
        TSL2561.Manual = False
        TSL2561.INTEG = TSL2561.Integration_Time.ms013
        TSL2561.Write(TSL2561.REG_ADDR.TIMING)

        TSL2561.THRESHLOW = &H11
        TSL2561.Write(TSL2561.REG_ADDR.THRESHLOW)

        TSL2561.THRESHHIGH = &H1234
        TSL2561.Write(TSL2561.REG_ADDR.THRESHHIGH)

        TSL2561.POWER = &H3
        TSL2561.Write(TSL2561.REG_ADDR.CONTROL)
    End Sub

    Private Sub btnStart_Click(sender As Object, e As RoutedEventArgs)
        RefreshTimer.Interval = TimeSpan.FromMilliseconds(100)
        RefreshTimer.Start()
    End Sub

    Private Sub Measurement(sender As Object, e As Object) Handles RefreshTimer.Tick
        RefreshTimer.Stop()

        TSL2561.Read(TSL2561.REG_ADDR.DATA0)
        Debug.WriteLine("DATA0 = {0:D}", TSL2561.DATA0)

        TSL2561.Read(TSL2561.REG_ADDR.DATA1)
        Debug.WriteLine("DATA1 = {0:D}", TSL2561.DATA1)

        TSL2561.Read(TSL2561.REG_ADDR.THRESHLOW)
    End Sub
End Class
```

```
Debug.WriteLine("THRESHLOW = {0:D}", TSL2561.THRESHLOW)

TSL2561.Read(TSL2561.REG_ADDR.THRESHHIGH)
Debug.WriteLine("THRESHHIGH = {0:D}", TSL2561.THRESHHIGH)

RefreshTimer.Start()
End Sub

Private Sub btnStop_Click(sender As Object, e As RoutedEventArgs)
    RefreshTimer.Stop()
End Sub

Private Sub btnBack_Click(sender As Object, e As RoutedEventArgs)
    RefreshTimer.Stop()
    TSL2561.close()
    App.rootFrame.Navigate(GetType(MenuPage))
End Sub
End Class
```

3-3.3 軸ジャイロセンサー ITG3200

変更履歴

2016年9月24日 第一版 新規作成

1. 概要

3 軸ジャイロ(X,Y,Z の角速度度)(ITG3200)と I2C 接続するための機能を提供します。

2. 機能一覧

プロパティ

| 機能名 | 機能説明 |
|---------------------|---|
| ID | デバイスのアイデンティティを検証するために使用されます。 |
| SMPLRT_DIV | ジャイロのサンプルレートを決定します。 |
| FS_SEL | ジャイロセンサデータのフルスケールの選択 |
| DLPF_CFG | デジタル低域通過フィルタの構成および内部サンプリングレートの設定 |
| ACTL | Logic level for INT output pin - 1=active low, 0=active high |
| OPEN | Drive type for INT output pin - 1=open drain, 0=push-pull |
| LATCH_INT_EN | Latch mode - 1=latch until interrupt is cleared, 0=50us pulse |
| INT_ANYRD_2CLEAR | Latch clear method - 1=any register read, 0=status register read only |
| TG_RDY_EN | デバイスは、(クロックソースを変更した後、PLL レディ)準備ができたときに割り込みを有効にします |
| RAW_RDY_EN | データが利用可能になったときに割り込みを有効にします |
| STATUS_ITG_RDY | 準備ができて PLL |
| STATUS_RAW_DATA_RDY | 生データは、準備ができています |

| 機能名 | 機能説明 |
|-----------------|--|
| TEMP_OUT | 16-bit temperature data |
| GYRO_XOUT | 16-bit X gyro output data |
| GYRO_YOUT | 16-bit Y gyro output data |
| GYRO_ZOUT | 16-bit Z gyro output data |
| PWR_MGM_H_RESET | Reset device and internal registers to the power-up-default settings |
| PWR_MGM_SLEEP | Enable low power sleep mode |
| PWR_MGM_STBY_XG | Put gyro X in standby mode (1=standby, 0=normal) |
| PWR_MGM_STBY_YG | Put gyro Y in standby mode (1=standby, 0=normal) |
| PWR_MGM_STBY_ZG | Put gyro Z in standby mode (1=standby, 0=normal) |
| PWR_MGM_CLK_SEL | Select device clock source |

メソッド

| 機能名 | 機能説明 |
|-------|---------------------------------------|
| Init | I2C バスコントローラに指定された接続設定を使って非同期的に接続します。 |
| Read | 指定レジスタ読込 |
| Write | 指定レジスタ書込 |

定数

| 機能名 | 機能説明 | 定数值 |
|---------------------|-----------------|------|
| REG_ADDR.WHO_AM_I | レジスタ WHO_AM_I | 0x00 |
| REG_ADDR.SMPLRT_DIV | レジスタ SMPLRT_DIV | 0x15 |
| REG_ADDR.DLPF_FS | レジスタ DLPF_FS | 0x16 |
| REG_ADDR.INT_CFG | レジスタ INT_CFG | 0x17 |
| REG_ADDR.INT_STATUS | レジスタ INT_STATUS | 0x1A |
| REG_ADDR.GYRO_DATA | レジスタ GYRO_XOUT | 0x1B |
| REG_ADDR.PWR_MGM | レジスタ PWR_MGM | 0x3E |

ID プロパティ

概要

WHO_AM_Iレジスタの ID の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.WHO_AM_I` を指定した時に保持されます。

Write メソッドで `REG_ADDR.WHO_AM_I` を指定した時に ID 値を使用して書込ます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.WHO_AM_I` を指定した時に保持した値
または、SET 処理をした値

SMPLRT_DIV プロパティ

概要

SMPLRT_DIV レジスタの SMPLRT_DIV の値を取得または設定します。

処理内容

Read メソッドで [REG_ADDR.SMPLRT_DIV](#) を指定した時に保持されます。

Write メソッドで [REG_ADDR.SMPLRT_DIV](#) を指定した時に SMPLRT_DIV 値を使用して書込まれます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで [REG_ADDR.SMPLRT_DIV](#) を指定した時に保持した値
または、SET 処理をした値

FS_SEL プロパティ

概要

DLPF_FS レジスタの FS_SEL の値を取得または設定します。

処理内容

Read メソッドで [REG_ADDR.DLPF_FS](#) を指定した時に保持されます。

Write メソッドで [REG_ADDR.DLPF_FS](#) を指定した時に FS_SEL 値を使用して書込ます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで [REG_ADDR.DLPF_FS](#) を指定した時に保持した値
または、SET 処理をした値

DLPF_CFG プロパティ

概要

DLPF_FS レジスタの DLPF_CFG の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.DLPF_FS` を指定した時に保持されます。

Write メソッドで `REG_ADDR.DLPF_FS` を指定した時に DLPF_CFG 値を使用して書込まず。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.DLPF_FS` を指定した時に保持した値
または、SET 処理をした値

ACTL プロパティ

概要

INT_CFG レジスタの ACTL の値を取得または設定します。

処理内容

Read メソッドで [REG_ADDR.INT_CFG](#) を指定した時に保持されます。

Write メソッドで [REG_ADDR.INT_CFG](#) を指定した時に ACTL 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで [REG_ADDR.INT_CFG](#) を指定した時に保持した値
または、SET 処理をした値

OPEN プロパティ

概要

INT_CFG レジスタの OPEN の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.INT_CFG` を指定した時に保持されます。

Write メソッドで `REG_ADDR.INT_CFG` を指定した時に OPEN 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.INT_CFG` を指定した時に保持した値
または、SET 処理をした値

LATCH_INT_EN プロパティ

概要

INT_CFG レジスタの LATCH_INT_EN の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.INT_CFG` を指定した時に保持されます。

Write メソッドで `REG_ADDR.INT_CFG` を指定した時に LATCH_INT_EN 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.INT_CFG` を指定した時に保持した値
または、SET 処理をした値

INT_ANYRD_2CLEAR プロパティ

概要

INT_CFG レジスタの INT_ANYRD_2CLEAR の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.INT_CFG` を指定した時に保持されます。

Write メソッドで `REG_ADDR.INT_CFG` を指定した時に INT_ANYRD_2CLEAR 値を使用して書込ます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.INT_CFG` を指定した時に保持した値
または、SET 処理をした値

TG_RDY_EN プロパティ

概要

INT_CFG レジスタの TG_RDY_EN の値を取得または設定します。

処理内容

Read メソッドで [REG_ADDR.INT_CFG](#) を指定した時に保持されます。

Write メソッドで [REG_ADDR.INT_CFG](#) を指定した時に TG_RDY_EN 値を使用して書込まず。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで [REG_ADDR.INT_CFG](#) を指定した時に保持した値
または、SET 処理をした値

RAW_RDY_EN プロパティ

概要

INT_CFG レジスタの RAW_RDY_EN の値を取得または設定します。

処理内容

Read メソッドで [REG_ADDR.INT_CFG](#) を指定した時に保持されます。

Write メソッドで [REG_ADDR.INT_CFG](#) を指定した時に RAW_RDY_EN 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで [REG_ADDR.INT_CFG](#) を指定した時に保持した値
または、SET 処理をした値

TG_RDY プロパティ

概要

INT_STATUS レジスタの ITG_RDY の値を取得または設定します。

処理内容

Read メソッドで [REG_ADDR.INT_STATUS](#) を指定した時に保持されます。

SET 処理

なし

GET 処理

Read メソッドで [REG_ADDR.INT_STATUS](#) を指定した時に保持した値

RAW_DATA_RDY プロパティ

概要

INT_STATUS レジスタの RAW_DATA_RDY の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.INT_STATUS` を指定した時に保持されます。

SET 処理

なし

GET 処理

Read メソッドで `REG_ADDR.INT_STATUS` を指定した時に保持した値

TEMP_OUT プロパティ

概要

TEMP_OUT_H、TEMP_OUT_L レジスタの値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.GYRO_DATA` を指定した時に保持されます。

SET 処理

なし

GET 処理

Read メソッドで `REG_ADDR.GYRO_DATA` を指定した時に保持した値
または、SET 処理をした値

GYRO_XOUT プロパティ

概要

GYRO_XOUT_H、GYRO_XOUT_L レジスタの値を取得または設定します。

処理内容

Read メソッドで [REG_ADDR.GYRO_DATA](#) を指定した時に保持されます。

SET 処理

なし

GET 処理

Read メソッドで [REG_ADDR.GYRO_DATA](#) を指定した時に保持した値

GYRO_YOUT プロパティ

概要

GYRO_YOUT_H、GYRO_YOUT_L レジスタの値を取得または設定します。

処理内容

Read メソッドで [REG_ADDR.GYRO_DATA](#) を指定した時に保持されます。

SET 処理

なし

GET 処理

Read メソッドで [REG_ADDR.GYRO_DATA](#) を指定した時に保持した値

GYRO_ZOUT プロパティ

概要

GYRO_ZOUT_H、GYRO_ZOUT_L レジスタの値を取得または設定します。

処理内容

Read メソッドで [REG_ADDR.GYRO_DATA](#) を指定した時に保持されます。

SET 処理

なし

GET 処理

Read メソッドで [REG_ADDR.GYRO_DATA](#) を指定した時に保持した値

H_RESET プロパティ

概要

PWR_MGM レジスタの H_RESET の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.PWR_MGM` を指定した時に保持されます。

Write メソッドで `REG_ADDR.PWR_MGM` を指定した時に H_RESET 値を使用して書込まず。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.PWR_MGM` を指定した時に保持した値
または、SET 処理をした値

SLEEP プロパティ

概要

PWR_MGM レジスタの SLEEP の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.PWR_MGM` を指定した時に保持されます。

Write メソッドで `REG_ADDR.PWR_MGM` を指定した時に SLEEP 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.PWR_MGM` を指定した時に保持した値
または、SET 処理をした値

STBY_XG プロパティ

概要

PWR_MGM レジスタの STBY_XG の値を取得または設定します。

処理内容

Read メソッドで [REG_ADDR.PWR_MGM](#) を指定した時に保持されます。

Write メソッドで [REG_ADDR.PWR_MGM](#) を指定した時に STBY_XG 値を使用して書込まず。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで [REG_ADDR.PWR_MGM](#) を指定した時に保持した値
または、SET 処理をした値

STBY_YG プロパティ

概要

PWR_MGM レジスタの STBY_YG の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.PWR_MGM` を指定した時に保持されます。

Write メソッドで `REG_ADDR.PWR_MGM` を指定した時に STBY_YG 値を使用して書込まず。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.PWR_MGM` を指定した時に保持した値
または、SET 処理をした値

STBY_ZG プロパティ

概要

PWR_MGM レジスタの STBY_ZG の値を取得または設定します。

処理内容

Read メソッドで [REG_ADDR.PWR_MGM](#) を指定した時に保持されます。

Write メソッドで [REG_ADDR.PWR_MGM](#) を指定した時に STBY_ZG 値を使用して書込まず。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで [REG_ADDR.PWR_MGM](#) を指定した時に保持した値
または、SET 処理をした値

CLK_SEL プロパティ

概要

PWR_MGM レジスタの CLK_SEL の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.PWR_MGM` を指定した時に保持されます。

Write メソッドで `REG_ADDR.PWR_MGM` を指定した時に CLK_SEL 値を使用して書込まず。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.PWR_MGM` を指定した時に保持した値
または、SET 処理をした値

ReadBuff プロパティ

概要

Read メソッド時の読込データをセットします。

処理内容

Read メソッド時に保持されます。

SET 処理

なし

GET 処理

Read メソッド時に保持した値

Init メソッド

概要

I2C バスコントローラに指定された接続設定を使って非同期的に接続します。

引数

| 引数名 | 型 | 説明 |
|---------|------|-----------|
| Address | byte | I2C のアドレス |

戻り値

なし

Read メソッド

概要

最新の変換結果を読込みます。

引数

| 引数名 | 型 | 説明 |
|----------|------|----|
| REG_ADDR | byte | |

戻り値

Status と同じ

1: 正常

1 以外: 異常

読込処理

| REG_ADDR 値 | 読込プロパティ |
|------------|------------------|
| WHO_AM_I | ID |
| SMPLRT_DIV | SMPLRT_DIV |
| DLPF_FS | FS_SEL |
| | DLPF_CFG |
| INT_CFG | ACTL |
| | OPEN |
| | LATCH_INT_EN |
| | INT_ANYRD_2CLEAR |
| | TG_RDY_EN |
| | RAW_RDY_EN |
| INT_STATUS | ITG_RDY |
| | RAW_DATA_RDY |
| GYRO_DATA | TEMP_OUT |
| | GYRO_XOUT |
| | GYRO_YOUT |
| | GYRO_ZOUT |

| REG_ADDR 値 | 読込プロパティ |
|------------|---------|
| PWR_MGM | H_RESET |
| | SLEEP |
| | STBY_XG |
| | STBY_YG |
| | STBY_ZG |
| | CLK_SEL |

Write メソッド

概要

引数 REG_ADDR に関するプロパティを編集し書き込みます。

引数

| 引数名 | 型 | 説明 |
|----------|------|----|
| REG_ADDR | byte | |

戻り値

Status

1: 正常

1 以外: 異常

書込処理

| REG_ADDR 値 | 書込プロパティ |
|------------|------------------|
| WHO_AM_I | ID |
| SMPLRT_DIV | SMPLRT_DIV |
| DLPF_FS | FS_SEL |
| | DLPF_CFG |
| INT_CFG | ACTL |
| | OPEN |
| | LATCH_INT_EN |
| | INT_ANYRD_2CLEAR |
| PWR_MGM | TG_RDY_EN |
| | RAW_RDY_EN |
| | H_RESET |
| PWR_MGM | SLEEP |
| | STBY_XG |
| | STBY_YG |
| PWR_MGM | STBY_ZG |
| | CLK_SEL |

サンプルプログラム

```
Imports I2CLibrary
Public NotInheritable Class ITG3200Page
    Inherits Page

    Private Const ITG3200_Address As Byte = &H77
    Private ITG3200 As ITG3200 = New ITG3200()
    Private WithEvents RefreshTimer As DispatcherTimer = New DispatcherTimer()

    Public Sub New()
        InitializeComponent()
        Call Init()
    End Sub

    Private Async Sub Init()
        Await ITG3200.Init(ITG3200_Address)

        ' DLPF_FS書込
        ITG3200.FS_SEL = &H3
        ITG3200.DLPF_CFG = &H0
        ITG3200.Write(ITG3200.REG_ADDR.DLPF_FS)

        ITG3200.Read(ITG3200.REG_ADDR.INT_CFG)
        If (ITG3200.Status = 1) Then
            Debug.WriteLine("ACTL = " & ITG3200.ACTL.ToString())
            Debug.WriteLine("OPEN = " & ITG3200.OPEN.ToString())
            Debug.WriteLine("LATCH_INT_EN = " & ITG3200.LATCH_INT_EN.ToString())
            Debug.WriteLine("INT_ANYRD_2CLEAR = " & ITG3200.INT_ANYRD_2CLEAR.ToString())
            Debug.WriteLine("TG_RDY_EN = " & ITG3200.TG_RDY_EN.ToString())
        End If

        ITG3200.Read(ITG3200.REG_ADDR.INT_STATUS)

        If (ITG3200.Status = 1) Then
            Debug.WriteLine("ITG_RDY = " & ITG3200.ITG_RDY.ToString())
            Debug.WriteLine("RAW_DATA_RDY = " & ITG3200.RAW_DATA_RDY.ToString())
        End If
    End Sub

    Private Sub btnStart_Click(sender As Object, e As RoutedEventArgs)
        RefreshTimer.Interval = TimeSpan.FromMilliseconds(100)
        RefreshTimer.Start()
    End Sub

    Private Sub Measurement(sender As Object, e As Object) Handles RefreshTimer.Tick
        RefreshTimer.Stop()

        ITG3200.Read(ITG3200.REG_ADDR.GYRO_DATA)
        If (ITG3200.Status = 1) Then
            Debug.WriteLine("TEMP_OUT = {0:D}", ITG3200.TEMP_OUT)
        End If
    End Sub
End Class
```

```
        Debug.WriteLine("GYRO_XOUT = {0:D}", ITG3200.GYRO_XOUT)
        Debug.WriteLine("GYRO_YOUT = {0:D}", ITG3200.GYRO_YOUT)
        Debug.WriteLine("GYRO_ZOUT = {0:D}", ITG3200.GYRO_ZOUT)
    End If

    RefreshTimer.Start()
End Sub

Private Sub btnStop_Click(sender As Object, e As RoutedEventArgs)
    RefreshTimer.Stop()
End Sub

Private Sub btnBack_Click(sender As Object, e As RoutedEventArgs)
    RefreshTimer.Stop()
    ITG3200.close()
    App.rootFrame.Navigate(GetType(MenuPage))
End Sub
End Class
```

3-4.タッチセンサー MPR121

変更履歴

2016年10月07日 第一版 新規作成

1. 概要

タッチセンサー(MPR121)とI2C 接続するための機能を提供します。

2. 機能一覧

プロパティ

| 機能名 | 機能説明 |
|-------|----------------------------|
| E0TS | ELE0 のタッチまたはリリース状態を取得します。 |
| E1TS | ELE1 のタッチまたはリリース状態を取得します。 |
| E2TS | ELE2 のタッチまたはリリース状態を取得します。 |
| E3TS | ELE3 のタッチまたはリリース状態を取得します。 |
| E4TS | ELE4 のタッチまたはリリース状態を取得します。 |
| E5TS | ELE5 のタッチまたはリリース状態を取得します。 |
| E6TS | ELE6 のタッチまたはリリース状態を取得します。 |
| E7TS | ELE7 のタッチまたはリリース状態を取得します。 |
| E8TS | ELE8 のタッチまたはリリース状態を取得します。 |
| E9TS | ELE9 のタッチまたはリリース状態を取得します。 |
| E10TS | ELE10 のタッチまたはリリース状態を取得します。 |
| E11TS | ELE11 のタッチまたはリリース状態を取得します。 |

| 機能名 | 機能説明 |
|--------|----------------------------------|
| E12TS | ELEPROX のタッチまたはリリース状態を取得します。 |
| OVCF | Over Current Flag を取得します。 |
| E0OOR | ELE0 の Out Of Range 状態を取得します。 |
| E1OOR | ELE1 の Out Of Range 状態を取得します。 |
| E2OOR | ELE2 の Out Of Range 状態を取得します。 |
| E3OOR | ELE3 の Out Of Range 状態を取得します。 |
| E4OOR | ELE4 の Out Of Range 状態を取得します。 |
| E5OOR | ELE5 の Out Of Range 状態を取得します。 |
| E6OOR | ELE6 の Out Of Range 状態を取得します。 |
| E7OOR | ELE7 の Out Of Range 状態を取得します。 |
| E8OOR | ELE8 の Out Of Range 状態を取得します。 |
| E9OOR | ELE9 の Out Of Range 状態を取得します。 |
| E10OOR | ELE10 の Out Of Range 状態を取得します。 |
| E11OOR | ELE11 の Out Of Range 状態を取得します。 |
| E12OOR | ELEPROX の Out Of Range 状態を取得します。 |
| ACFF | AUTO-CONFIG Fail Flag を取得します。 |
| ARFF | AUTO-RECONFIG Fail Flag を取得します。 |

| 機能名 | 機能説明 |
|-------|---|
| E0FD | ELE0 の Electrode Filtered Data を取得します。 |
| E1FD | ELE1 の Electrode Filtered Data を取得します。 |
| E2FD | ELE2 の Electrode Filtered Data を取得します。 |
| E3FD | ELE3 の Electrode Filtered Data を取得します。 |
| E4FD | ELE4 の Electrode Filtered Data を取得します。 |
| E5FD | ELE5 の Electrode Filtered Data を取得します。 |
| E6FD | ELE6 の Electrode Filtered Data を取得します。 |
| E7FD | ELE7 の Electrode Filtered Data を取得します。 |
| E8FD | ELE8 の Electrode Filtered Data を取得します。 |
| E9FD | ELE9 の Electrode Filtered Data を取得します。 |
| E10FD | ELE10 の Electrode Filtered Data を取得します。 |
| E11FD | ELE11 の Electrode Filtered Data を取得します。 |
| E12FD | ELEPROX の Electrode Filtered Data を取得します。 |
| E0BV | ELE0 の Baseline Value を取得します。 |
| E1BV | ELE1 の Baseline Value を取得します。 |
| E2BV | ELE2 の Baseline Value を取得します。 |
| E3BV | ELE3 の Baseline Value を取得します。 |

| 機能名 | 機能説明 |
|-------|---|
| E4BV | ELE4 の Baseline Value を取得します。 |
| E5BV | ELE5 の Baseline Value を取得します。 |
| E6BV | ELE6 の Baseline Value を取得します。 |
| E7BV | ELE7 の Baseline Value を取得します。 |
| E8BV | ELE8 の Baseline Value を取得します。 |
| E9BV | ELE9 の Baseline Value を取得します。 |
| E10BV | ELE10 の Baseline Value を取得します。 |
| E11BV | ELE11 の Baseline Value を取得します。 |
| E12BV | ELEPROX の Baseline Value を取得します。 |
| MHDR | Maximum Half Delta Rising を取得または設定します。 |
| NHDR | Noise Half Delta Rising を取得または設定します。 |
| NCLR | Noise Count Limit Rising を取得または設定します。 |
| FDLR | Filter Delay Count Limit Rising を取得または設定します。 |
| MHDF | Maximum Half Delta Falling を取得または設定します。 |
| NHDF | Noise Half Delta Falling を取得または設定します。 |
| NCLF | Noise Count Limit Falling を取得または設定します。 |
| FDLF | Filter Delay Count Limit Falling を取得または設定します。 |
| NHDT | Maximum Half Delta Touched を取得または設定します。 |

| 機能名 | 機能説明 |
|----------|---|
| NCLT | Noise Count Limit Touched を取得または設定します。 |
| FDLT | Filter Delay Count Limit Touched を取得または設定します。 |
| MHDPROXR | ELEPROX の Maximum Half Delta Rising を取得または設定します。 |
| NHDPROXR | ELEPROX の Noise Half Delta Rising を取得または設定します。 |
| NCLPROXR | ELEPROX の Noise Count Limit Rising を取得または設定します。 |
| FDLPROXR | ELEPROX の Filter Delay Count Limit Rising を取得または設定します。 |
| MHDPROXF | ELEPROX の Maximum Half Delta Falling を取得または設定します。 |
| NHDPROXF | ELEPROX の Noise Half Delta Falling を取得または設定します。 |
| NCLPROXF | ELEPROX の Noise Count Limit Falling を取得または設定します。 |
| FDLPROXF | ELEPROX の Filter Delay Count Limit Falling を取得または設定します。 |
| NHDPROXT | ELEPROX の Noise Half Delta Touched を取得または設定します。 |
| NCLPROXT | ELEPROX の Noise Count Limit Touched を取得または設定します。 |
| FDLPROXT | ELEPROX の Filter Delay Count Limit Touched を取得または設定します。 |
| E0TTH | ELE0 のタッチ閾値を取得または設定します。 |
| E0RTH | ELE0 のリリース閾値を取得または設定します。 |
| E1TTH | ELE1 のタッチ閾値を取得または設定します。 |
| E1RTH | ELE1 のリリース閾値を取得または設定します。 |
| E2TTH | ELE2 のタッチ閾値を取得または設定します。 |

| 機能名 | 機能説明 |
|--------|---------------------------|
| E2RTH | ELE2 のリリース閾値を取得または設定します。 |
| E3TTH | ELE3 のタッチ閾値を取得または設定します。 |
| E3RTH | ELE3 のリリース閾値を取得または設定します。 |
| E4TTH | ELE4 のタッチ閾値を取得または設定します。 |
| E4RTH | ELE4 のリリース閾値を取得または設定します。 |
| E5TTH | ELE5 のタッチ閾値を取得または設定します。 |
| E5RTH | ELE5 のリリース閾値を取得または設定します。 |
| E6TTH | ELE6 のタッチ閾値を取得または設定します。 |
| E6RTH | ELE6 のリリース閾値を取得または設定します。 |
| E7TTH | ELE7 のタッチ閾値を取得または設定します。 |
| E7RTH | ELE7 のリリース閾値を取得または設定します。 |
| E8TTH | ELE8 のタッチ閾値を取得または設定します。 |
| E8RTH | ELE8 のリリース閾値を取得または設定します。 |
| E9TTH | ELE9 のタッチ閾値を取得または設定します。 |
| E9RTH | ELE9 のリリース閾値を取得または設定します。 |
| E10TTH | ELE10 のタッチ閾値を取得または設定します。 |
| E10RTH | ELE10 のリリース閾値を取得または設定します。 |
| E11TTH | ELE11 のタッチ閾値を取得または設定します。 |

| 機能名 | 機能説明 |
|---------|--|
| E11RTH | ELE11 のリリース閾値を取得または設定します。 |
| E12TTH | ELEPROX のタッチ閾値を取得または設定します。 |
| E12RTH | ELEPROX のリリース閾値を取得または設定します。 |
| DR | リリース用のデバウンス数を取得または設定します。 |
| DT | タッチ用のデバウンス数を取得または設定します。 |
| FFI | First Filter Iterations を取得または設定します。 |
| CDC | Charge Discharge Current を取得または設定します。 |
| CDT | Charge Discharge Time を取得または設定します。 |
| SFI | Second Filter Iterations を取得または設定します。 |
| ESI | Electrode Sample Interval を取得または設定します。 |
| CL | Calibration Lock bit を取得または設定します。 |
| ELEPROX | ELEPROX ピンの組み合わせを取得または設定します。 |
| ELE | 各ピンの電極タッチ検出の有効／無効を取得または設定します。 |
| CDC0 | ELE0 の Electrode Current を取得または設定します。 |
| CDC1 | ELE1 の Electrode Current を取得または設定します。 |
| CDC2 | ELE2 の Electrode Current を取得または設定します。 |
| CDC3 | ELE3 の Electrode Current を取得または設定します。 |
| CDC4 | ELE4 の Electrode Current を取得または設定します。 |

| 機能名 | 機能説明 |
|-------|--|
| CDC5 | ELE5 の Electrode Current を取得または設定します。 |
| CDC6 | ELE6 の Electrode Current を取得または設定します。 |
| CDC7 | ELE7 の Electrode Current を取得または設定します。 |
| CDC8 | ELE8 の Electrode Current を取得または設定します。 |
| CDC9 | ELE9 の Electrode Current を取得または設定します。 |
| CDC10 | ELE10 の Electrode Current を取得または設定します。 |
| CDC11 | ELE11 の Electrode Current を取得または設定します。 |
| CDC12 | ELEPROX の Electrode Current を取得または設定します。 |
| CDT0 | ELE0 の Charge Time を取得または設定します。 |
| CDT1 | ELE1 の Charge Time を取得または設定します。 |
| CDT2 | ELE2 の Charge Time を取得または設定します。 |
| CDT3 | ELE3 の Charge Time を取得または設定します。 |
| CDT4 | ELE4 の Charge Time を取得または設定します。 |
| CDT5 | ELE5 の Charge Time を取得または設定します。 |
| CDT6 | ELE6 の Charge Time を取得または設定します。 |
| CDT7 | ELE7 の Charge Time を取得または設定します。 |
| CDT8 | ELE8 の Charge Time を取得または設定します。 |
| CDT9 | ELE9 の Charge Time を取得または設定します。 |

| 機能名 | 機能説明 |
|-------|---|
| CDT10 | ELE10 の Charge Time を取得または設定します。 |
| CDT11 | ELE11 の Charge Time を取得または設定します。 |
| CDT12 | ELEPROX の Charge Time を取得または設定します。 |
| CTL0 | GPIO Control Register 0 を取得または設定します。 |
| CTL1 | GPIO Control Register 1 を取得または設定します。 |
| DAT | GPIO Data Register を取得または設定します。 |
| DIR | GPIO Direction Control Register を取得または設定します。 |
| EN | GPIO Enable Register を取得または設定します。 |
| SET | GPIO Data Set Register を取得または設定します。 |
| CLR | GPIO Data Clear Register を取得または設定します。 |
| TOG | GPIO Data Toggle Register を取得または設定します。 |
| AFES | AFE Samples during binary search を取得または設定します。 |
| RETRY | MPR121 may retry AUTO-CONFIG or AUTO-RECONFIG after failed を取得または設定します。 |
| BVA | Baseline Value Adjust Control を取得または設定します。 |
| ARE | AUTO-RECONFIG の有効／無効を取得または設定します。 |
| ACE | AUTO-CONFIG の有効／無効を取得または設定します。 |
| SCTS | Skip Charge Time Search を取得または設定します。 |
| OORIE | Out of Range Interrupt Enable を取得または設定します。 |

| 機能名 | 機能説明 |
|-------|--|
| ARFIE | AUTO-RECONFIG Fail Interrupt Enable を取得または設定します。 |
| ACFIE | AUTO-CONFIG Fail Interrupt Enable を取得または設定します。 |
| USL | AUTO-CONFIG USL Register を取得または設定します。 |
| LSL | Upper Side Limit.を取得または設定します。 |
| TL | Lower Side Limit.を取得または設定します。 |
| SRST | 電源オン後の状態にリセットします。 |

メソッド

| 機能名 | 機能説明 |
|-------|---------------------------------------|
| Init | I2C バスコントローラに指定された接続設定を使って非同期的に接続します。 |
| Read | 指定レジスタ読込 |
| Write | 指定レジスタ書込 |

定数

| 機能名 | 機能説明 | 定数值 |
|----------------------------------|--------------------------------------|------|
| REG_ADDR.Touch_Status | レジスタ Touch_Status | 0x00 |
| REG_ADDR.Out_Of_Range_Status | レジスタ OOR Status | 0x02 |
| REG_ADDR.Electrode_Data | レジスタ Electrode Filtered Data | 0x04 |
| REG_ADDR.Baseline_Value | レジスタ Baseline Value | 0x1E |
| REG_ADDR.Rising | レジスタ MHD Rising | 0x2B |
| REG_ADDR.Falling | レジスタ MHD Falling | 0x2F |
| REG_ADDR.Touched | レジスタ NHD Touched | 0x33 |
| REG_ADDR.ELEPROX_Rising | レジスタ ELEPROX MHD Rising | 0x36 |
| REG_ADDR.ELEPROX_Falling | レジスタ ELEPROX MHD Falling | 0x3A |
| REG_ADDR.ELEPROX_Touched | レジスタ ELEPROX NHD Touched | 0x3E |
| REG_ADDR.Threshold | レジスタ ELE0 Touch Threshold | 0x41 |
| REG_ADDR.Debounce | レジスタ Debounce Touch & Release | 0x5B |
| REG_ADDR.AF_Configuration_1 | レジスタ AFE Configuration 1 | 0x5C |
| REG_ADDR.AF_Configuration_2 | レジスタ AFE Configuration 2 | 0x5D |
| REG_ADDR.Electrode_Configuration | レジスタ Electrode Configuration(ECR) | 0x5E |
| REG_ADDR.Electrode_Current | レジスタ ELE0 Electrode Current | 0x5F |
| REG_ADDR.Charge_Time | レジスタ ELE0, ELE1 Charge Time | 0x6C |
| REG_ADDR.GPIO | レジスタ GPIO Control Register 0 | 0x73 |
| REG_ADDR.AUTO_CONFIG | AUTO-CONFIG Control Register 0 | 0x7B |
| REG_ADDR.Soft_Reset | Soft Reset Register | 0x80 |

E0TS プロパティ

概要

ELE0 のタッチまたはリリース状態を取得します。

処理内容

Read メソッドで `REG_ADDR.Touch_Status` を指定した時に ELE0 のタッチ状態が保持されます。

1,タッチ状態

0,リリース状態

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Touch_Status` を指定した時に保持した値
または、SET 処理をした値

E1TS プロパティ

概要

ELE1 のタッチまたはリリース状態を取得します。

処理内容

Read メソッドで `REG_ADDR.Touch_Status` を指定した時に ELE1 のタッチ状態が保持されます。

1,タッチ状態

0,リリース状態

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Touch_Status` を指定した時に保持した値
または、SET 処理をした値

E2TS プロパティ

概要

ELE2 のタッチまたはリリース状態を取得します。

処理内容

Read メソッドで `REG_ADDR.Touch_Status` を指定した時に ELE2 のタッチ状態が保持されます。

1, タッチ状態

0, リリース状態

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Touch_Status` を指定した時に保持した値
または、SET 処理をした値

E3TS プロパティ

概要

ELE3 のタッチまたはリリース状態を取得します。

処理内容

Read メソッドで `REG_ADDR.Touch_Status` を指定した時に ELE3 のタッチ状態が保持されます。

1,タッチ状態

0,リリース状態

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Touch_Status` を指定した時に保持した値
または、SET 処理をした値

E4TS プロパティ

概要

ELE4 のタッチまたはリリース状態を取得します。

処理内容

Read メソッドで `REG_ADDR.Touch_Status` を指定した時に ELE4 のタッチ状態が保持されます。

1,タッチ状態

0,リリース状態

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Touch_Status` を指定した時に保持した値
または、SET 処理をした値

E5TS プロパティ

概要

ELE5 のタッチまたはリリース状態を取得します。

処理内容

Read メソッドで `REG_ADDR.Touch_Status` を指定した時に ELE5 のタッチ状態が保持されます。

1,タッチ状態

0,リリース状態

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Touch_Status` を指定した時に保持した値
または、SET 処理をした値

E6TS プロパティ

概要

ELE6 のタッチまたはリリース状態を取得します。

処理内容

Read メソッドで `REG_ADDR.Touch_Status` を指定した時に ELE6 のタッチ状態が保持されます。

1,タッチ状態

0,リリース状態

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Touch_Status` を指定した時に保持した値
または、SET 処理をした値

E7TS プロパティ

概要

ELE7 のタッチまたはリリース状態を取得します。

処理内容

Read メソッドで `REG_ADDR.Touch_Status` を指定した時に ELE7 のタッチ状態が保持されます。

1,タッチ状態

0,リリース状態

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Touch_Status` を指定した時に保持した値
または、SET 処理をした値

E8TS プロパティ

概要

ELE8 のタッチまたはリリース状態を取得します。

処理内容

Read メソッドで `REG_ADDR.Touch_Status` を指定した時に ELE8 のタッチ状態が保持されます。

1,タッチ状態

0,リリース状態

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Touch_Status` を指定した時に保持した値
または、SET 処理をした値

E9TS プロパティ

概要

ELE9 のタッチまたはリリース状態を取得します。

処理内容

Read メソッドで `REG_ADDR.Touch_Status` を指定した時に ELE9 のタッチ状態が保持されます。

1,タッチ状態

0,リリース状態

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Touch_Status` を指定した時に保持した値
または、SET 処理をした値

E10TS プロパティ

概要

ELE10 のタッチまたはリリース状態を取得します。

処理内容

Read メソッドで `REG_ADDR.Touch_Status` を指定した時に ELE10 のタッチ状態が保持されます。

1,タッチ状態

0,リリース状態

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Touch_Status` を指定した時に保持した値
または、SET 処理をした値

E11TS プロパティ

概要

ELE11 のタッチまたはリリース状態を取得します。

処理内容

Read メソッドで `REG_ADDR.Touch_Status` を指定した時に ELE11 のタッチ状態が保持されます。

1,タッチ状態

0,リリース状態

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Touch_Status` を指定した時に保持した値
または、SET 処理をした値

E12TS プロパティ

概要

ELEPROX のタッチまたはリリース状態を取得します。

処理内容

Read メソッドで `REG_ADDR.Touch_Status` を指定した時に ELEPROX のタッチ状態が保持されます。

1,タッチ状態

0,リリース状態

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Touch_Status` を指定した時に保持した値
または、SET 処理をした値

OVCF プロパティ

概要

Over Current Flag を取得します。

処理内容

Read メソッドで `REG_ADDR.Touch_Status` を指定した時に Over Current Flag(過電流フラグ)が保持されます。

1,REXT ピンで過電流検出

0,通常状態

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Touch_Status` を指定した時に保持した値
または、SET 処理をした値

E00OR プロパティ

概要

ELE0 の Out Of Range 状態を取得します。

処理内容

Read メソッドで `REG_ADDR.Out_Of_Range_Status` を指定した時に Out Of Range Status が保持されます。

1,Out Of Range エラー

0,通常状態

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Out_Of_Range_Status` を指定した時に保持した値
または、SET 処理をした値

E10OR プロパティ

概要

ELE1 の Out Of Range 状態を取得します。

処理内容

Read メソッドで `REG_ADDR.Out_Of_Range_Status` を指定した時に Out Of Range Status が保持されます。

1, Out Of Range エラー

0, 通常状態

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Out_Of_Range_Status` を指定した時に保持した値
または、SET 処理をした値

E2OOR プロパティ

概要

ELE2 の Out Of Range 状態を取得します。

処理内容

Read メソッドで `REG_ADDR.Out_Of_Range_Status` を指定した時に Out Of Range Status が保持されます。

1,Out Of Range エラー

0,通常状態

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Out_Of_Range_Status` を指定した時に保持した値
または、SET 処理をした値

E30OR プロパティ

概要

ELE3 の Out Of Range 状態を取得します。

処理内容

Read メソッドで `REG_ADDR.Out_Of_Range_Status` を指定した時に Out Of Range Status が保持されます。

1,Out Of Range エラー

0,通常状態

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Out_Of_Range_Status` を指定した時に保持した値
または、SET 処理をした値

E4OOR プロパティ

概要

ELE4 の Out Of Range 状態を取得します。

処理内容

Read メソッドで `REG_ADDR.Out_Of_Range_Status` を指定した時に Out Of Range Status が保持されます。

1, Out Of Range エラー

0, 通常状態

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Out_Of_Range_Status` を指定した時に保持した値
または、SET 処理をした値

E50OR プロパティ

概要

ELE5 の Out Of Range 状態を取得します。

処理内容

Read メソッドで `REG_ADDR.Out_Of_Range_Status` を指定した時に Out Of Range Status が保持されます。

1,Out Of Range エラー

0,通常状態

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Out_Of_Range_Status` を指定した時に保持した値
または、SET 処理をした値

E60OR プロパティ

概要

ELE6 の Out Of Range 状態を取得します。

処理内容

Read メソッドで `REG_ADDR.Out_Of_Range_Status` を指定した時に Out Of Range Status が保持されます。

1,Out Of Range エラー

0,通常状態

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Out_Of_Range_Status` を指定した時に保持した値
または、SET 処理をした値

E70OR プロパティ

概要

ELE7 の Out Of Range 状態を取得します。

処理内容

Read メソッドで `REG_ADDR.Out_Of_Range_Status` を指定した時に Out Of Range Status が保持されます。

1,Out Of Range エラー

0,通常状態

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Out_Of_Range_Status` を指定した時に保持した値
または、SET 処理をした値

E80OR プロパティ

概要

ELE8 の Out Of Range 状態を取得します。

処理内容

Read メソッドで `REG_ADDR.Out_Of_Range_Status` を指定した時に Out Of Range Status が保持されます。

1,Out Of Range エラー

0,通常状態

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Out_Of_Range_Status` を指定した時に保持した値
または、SET 処理をした値

E90OR プロパティ

概要

ELE9 の Out Of Range 状態を取得します。

処理内容

Read メソッドで `REG_ADDR.Out_Of_Range_Status` を指定した時に Out Of Range Status が保持されます。

1, Out Of Range エラー

0, 通常状態

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Out_Of_Range_Status` を指定した時に保持した値
または、SET 処理をした値

E100OR プロパティ

概要

ELE10 の Out Of Range 状態を取得します。

処理内容

Read メソッドで `REG_ADDR.Out_Of_Range_Status` を指定した時に Out Of Range Status が保持されます。

1,Out Of Range エラー

0,通常状態

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Out_Of_Range_Status` を指定した時に保持した値
または、SET 処理をした値

E110OR プロパティ

概要

ELE11 の Out Of Range 状態を取得します。

処理内容

Read メソッドで `REG_ADDR.Out_Of_Range_Status` を指定した時に Out Of Range Status が保持されます。

1,Out Of Range エラー

0,通常状態

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Out_Of_Range_Status` を指定した時に保持した値
または、SET 処理をした値

E120OR プロパティ

概要

ELEPROX の Out Of Range 状態を取得します。

処理内容

Read メソッドで `REG_ADDR.Out_Of_Range_Status` を指定した時に Out Of Range Status が保持されます。

1,Out Of Range エラー

0,通常状態

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Out_Of_Range_Status` を指定した時に保持した値
または、SET 処理をした値

ACFF プロパティ

概要

AUTO-CONFIG Fail Flag を取得します。

処理内容

Read メソッドで `REG_ADDR.Out_Of_Range_Status` を指定した時に AUTO-CONFIG Fail Flag が保持されます。

AUTO-CONFIG が失敗した場合このビットがセットされます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Out_Of_Range_Status` を指定した時に保持した値
または、SET 処理をした値

ARFF プロパティ

概要

AUTO-RECONFIG Fail Flag を取得します。

処理内容

Read メソッドで `REG_ADDR.Out_Of_Range_Status` を指定した時に AUTO-RECONFIG Fail Flag が保持されます。

AUTO-RECONFIG が失敗した場合このビットがセットされます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Out_Of_Range_Status` を指定した時に保持した値
または、SET 処理をした値

E0FD プロパティ

概要

ELE0 の Electrode Filtered Data を取得します。

処理内容

Read メソッドで `REG_ADDR.Electrode_Data` を指定した時に Electrode Filtered Data が保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Electrode_Data` を指定した時に保持した値
または、SET 処理をした値

E1FD プロパティ

概要

ELE1 の Electrode Filtered Data を取得します。

処理内容

Read メソッドで `REG_ADDR.Electrode_Data` を指定した時に Electrode Filtered Data が保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Electrode_Data` を指定した時に保持した値
または、SET 処理をした値

E2FD プロパティ

概要

ELE2 の Electrode Filtered Data を取得します。

処理内容

Read メソッドで `REG_ADDR.Electrode_Data` を指定した時に Electrode Filtered Data が保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Electrode_Data` を指定した時に保持した値
または、SET 処理をした値

E3FD プロパティ

概要

ELE3 の Electrode Filtered Data を取得します。

処理内容

Read メソッドで `REG_ADDR.Electrode_Data` を指定した時に Electrode Filtered Data が保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Electrode_Data` を指定した時に保持した値
または、SET 処理をした値

E4FD プロパティ

概要

ELE4 の Electrode Filtered Data を取得します。

処理内容

Read メソッドで `REG_ADDR.Electrode_Data` を指定した時に Electrode Filtered Data が保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Electrode_Data` を指定した時に保持した値
または、SET 処理をした値

E5FD プロパティ

概要

ELE5 の Electrode Filtered Data を取得します。

処理内容

Read メソッドで `REG_ADDR.Electrode_Data` を指定した時に Electrode Filtered Data が保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Electrode_Data` を指定した時に保持した値
または、SET 処理をした値

E6FD プロパティ

概要

ELE6 の Electrode Filtered Data を取得します。

処理内容

Read メソッドで `REG_ADDR.Electrode_Data` を指定した時に Electrode Filtered Data が保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Electrode_Data` を指定した時に保持した値
または、SET 処理をした値

E7FD プロパティ

概要

ELE7 の Electrode Filtered Data を取得します。

処理内容

Read メソッドで `REG_ADDR.Electrode_Data` を指定した時に Electrode Filtered Data が保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Electrode_Data` を指定した時に保持した値
または、SET 処理をした値

E8FD プロパティ

概要

ELE8 の Electrode Filtered Data を取得します。

処理内容

Read メソッドで `REG_ADDR.Electrode_Data` を指定した時に Electrode Filtered Data が保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Electrode_Data` を指定した時に保持した値
または、SET 処理をした値

E9FD プロパティ

概要

ELE9 の Electrode Filtered Data を取得します。

処理内容

Read メソッドで `REG_ADDR.Electrode_Data` を指定した時に Electrode Filtered Data が保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Electrode_Data` を指定した時に保持した値
または、SET 処理をした値

E10FD プロパティ

概要

ELE10 の Electrode Filtered Data を取得します。

処理内容

Read メソッドで `REG_ADDR.Electrode_Data` を指定した時に Electrode Filtered Data が保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Electrode_Data` を指定した時に保持した値
または、SET 処理をした値

E11FD プロパティ

概要

ELE11 の Electrode Filtered Data を取得します。

処理内容

Read メソッドで `REG_ADDR.Electrode_Data` を指定した時に Electrode Filtered Data が保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Electrode_Data` を指定した時に保持した値
または、SET 処理をした値

E12FD プロパティ

概要

ELEPROX の Electrode Filtered Data を取得します。

処理内容

Read メソッドで `REG_ADDR.Electrode_Data` を指定した時に Electrode Filtered Data が保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Electrode_Data` を指定した時に保持した値
または、SET 処理をした値

E0BV プロパティ

概要

ELE0 の Baseline Value を取得します。

処理内容

Read メソッドで `REG_ADDR.Baseline_Value` を指定した時に Baseline Value が保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Baseline_Value` を指定した時に保持した値
または、SET 処理をした値

E1BV プロパティ

概要

ELE1 の Baseline Value を取得します。

処理内容

Read メソッドで `REG_ADDR.Baseline_Value` を指定した時に Baseline Value が保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Baseline_Value` を指定した時に保持した値
または、SET 処理をした値

E2BV プロパティ

概要

ELE2 の Baseline Value を取得します。

処理内容

Read メソッドで `REG_ADDR.Baseline_Value` を指定した時に Baseline Value が保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Baseline_Value` を指定した時に保持した値
または、SET 処理をした値

E3BV プロパティ

概要

ELE3 の Baseline Value を取得します。

処理内容

Read メソッドで `REG_ADDR.Baseline_Value` を指定した時に Baseline Value が保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Baseline_Value` を指定した時に保持した値
または、SET 処理をした値

E4BV プロパティ

概要

ELE4 の Baseline Value を取得します。

処理内容

Read メソッドで `REG_ADDR.Baseline_Value` を指定した時に Baseline Value が保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Baseline_Value` を指定した時に保持した値
または、SET 処理をした値

E5BV プロパティ

概要

ELE5 の Baseline Value を取得します。

処理内容

Read メソッドで `REG_ADDR.Baseline_Value` を指定した時に Baseline Value が保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Baseline_Value` を指定した時に保持した値
または、SET 処理をした値

E6BV プロパティ

概要

ELE6 の Baseline Value を取得します。

処理内容

Read メソッドで `REG_ADDR.Baseline_Value` を指定した時に Baseline Value が保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Baseline_Value` を指定した時に保持した値
または、SET 処理をした値

E7BV プロパティ

概要

ELE7 の Baseline Value を取得します。

処理内容

Read メソッドで `REG_ADDR.Baseline_Value` を指定した時に Baseline Value が保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Baseline_Value` を指定した時に保持した値
または、SET 処理をした値

E8BV プロパティ

概要

ELE8 の Baseline Value を取得します。

処理内容

Read メソッドで `REG_ADDR.Baseline_Value` を指定した時に Baseline Value が保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Baseline_Value` を指定した時に保持した値
または、SET 処理をした値

E9BV プロパティ

概要

ELE9 の Baseline Value を取得します。

処理内容

Read メソッドで `REG_ADDR.Baseline_Value` を指定した時に Baseline Value が保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Baseline_Value` を指定した時に保持した値
または、SET 処理をした値

E10BV プロパティ

概要

ELE10 の Baseline Value を取得します。

処理内容

Read メソッドで `REG_ADDR.Baseline_Value` を指定した時に Baseline Value が保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Baseline_Value` を指定した時に保持した値
または、SET 処理をした値

E11BV プロパティ

概要

ELE11 の Baseline Value を取得します。

処理内容

Read メソッドで `REG_ADDR.Baseline_Value` を指定した時に Baseline Value が保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Baseline_Value` を指定した時に保持した値
または、SET 処理をした値

E12BV プロパティ

概要

ELEPROX の Baseline Value を取得します。

処理内容

Read メソッドで `REG_ADDR.Baseline_Value` を指定した時に Baseline Value が保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Baseline_Value` を指定した時に保持した値
または、SET 処理をした値

MHDR プロパティ

概要

Maximum Half Delta Rising を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Rising` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Rising` を指定した時に MDHR 値を使用して書込みます。

有効値の範囲は 1～63 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Rising` を指定した時に保持した値

または、SET 処理をした値

NHDR プロパティ

概要

Noise Half Delta Rising を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Rising` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Rising` を指定した時に NHDR 値を使用して書込ます。

有効値の範囲は 1～63 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Rising` を指定した時に保持した値

または、SET 処理をした値

NCLR プロパティ

概要

Noise Count Limit Rising を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Rising` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Rising` を指定した時に NCLR 値を使用して書込みます。

有効値の範囲は 0~255 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Rising` を指定した時に保持した値

または、SET 処理をした値

FDLR プロパティ

概要

Filter Delay Count Limit Rising を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Rising` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Rising` を指定した時に FDLR 値を使用して書込みます。

有効値の範囲は 0~255 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Rising` を指定した時に保持した値

または、SET 処理をした値

MHDF プロパティ

概要

Maximum Half Delta Falling を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Falling` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Falling` を指定した時に MHDF 値を使用して書込みます。

有効値の範囲は 1～63 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Falling` を指定した時に保持した値

または、SET 処理をした値

NHDF プロパティ

概要

Noise Half Delta Falling を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Falling` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Falling` を指定した時に NHDF 値を使用して書込みます。

有効値の範囲は 1～63 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Falling` を指定した時に保持した値

または、SET 処理をした値

NCLF プロパティ

概要

Noise Count Limit Falling を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Falling` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Falling` を指定した時に NCLF 値を使用して書込みます。

有効値の範囲は 0~255 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Falling` を指定した時に保持した値

または、SET 処理をした値

FDLF プロパティ

概要

Filter Delay Count Limit Falling を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Falling` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Falling` を指定した時に FDLF 値を使用して書込みます。

有効値の範囲は 0~255 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Falling` を指定した時に保持した値

または、SET 処理をした値

NHDT プロパティ

概要

Maximum Half Delta Touched を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Touched` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Touched` を指定した時に NHDT 値を使用して書込みます。

有効値の範囲は 1～63 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Touched` を指定した時に保持した値

または、SET 処理をした値

NCLT プロパティ

概要

Noise Half Delta Touched を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Touched` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Touched` を指定した時に NCLT 値を使用して書込みます。

有効値の範囲は 0~255 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Touched` を指定した時に保持した値
または、SET 処理をした値

FDLT プロパティ

概要

Filter Delay Count Limit Touched を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Touched` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Touched` を指定した時に FDLT 値を使用して書込みます。

有効値の範囲は 0~255 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Touched` を指定した時に保持した値

または、SET 処理をした値

FDLT プロパティ

概要

Filter Delay Count Limit Touched を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Touched` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Touched` を指定した時に FDLT 値を使用して書込みます。

有効値の範囲は 0~255 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Touched` を指定した時に保持した値
または、SET 処理をした値

MHDPROXR プロパティ

概要

ELEPROX の Maximum Half Delta Rising を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.ELEPROX_Rising` を指定した時に保持されます。

Write メソッドで `REG_ADDR.ELEPROX_Rising` を指定した時に MHDPROXR 値を使用して書込ます。

有効値の範囲は 1～63 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.ELEPROX_Rising` を指定した時に保持した値
または、SET 処理をした値

NHDPROXR プロパティ

概要

ELEPROX の Noise Half Delta Rising を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.ELEPROX_Rising` を指定した時に保持されます。

Write メソッドで `REG_ADDR.ELEPROX_Rising` を指定した時に NHDPROXR 値を使用して書込みます。

有効値の範囲は 1～63 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.ELEPROX_Rising` を指定した時に保持した値
または、SET 処理をした値

NCLPROXR プロパティ

概要

ELEPROX の Noise Count Limit Rising を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.ELEPROX_Rising` を指定した時に保持されます。

Write メソッドで `REG_ADDR.ELEPROX_Rising` を指定した時に NCLPROXR 値を使用して書込みます。

有効値の範囲は 0~255 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.ELEPROX_Rising` を指定した時に保持した値
または、SET 処理をした値

FDLPROXR プロパティ

概要

ELEPROX の Filter Delay Count Limit Rising を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.ELEPROX_Rising` を指定した時に保持されます。

Write メソッドで `REG_ADDR.ELEPROX_Rising` を指定した時に FDLPROXR 値を使用して書込みます。

有効値の範囲は 0~255 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.ELEPROX_Rising` を指定した時に保持した値
または、SET 処理をした値

MHDPROXF プロパティ

概要

ELEPROX の Maximum Half Delta Falling を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.ELEPROX_Falling` を指定した時に保持されます。

Write メソッドで `REG_ADDR.ELEPROX_Falling` を指定した時に MHDPROXF 値を使用して書込みます。

有効値の範囲は 1～63 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.ELEPROX_Falling` を指定した時に保持した値
または、SET 処理をした値

NHDPROXF プロパティ

概要

ELEPROX の Noise Half Delta Falling を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.ELEPROX_Falling` を指定した時に保持されます。

Write メソッドで `REG_ADDR.ELEPROX_Falling` を指定した時に NHDPROXF 値を使用して書込ます。

有効値の範囲は 1～63 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.ELEPROX_Falling` を指定した時に保持した値
または、SET 処理をした値

NCLPROXF プロパティ

概要

ELEPROX の Noise Count Limit Falling を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.ELEPROX_Falling` を指定した時に保持されます。

Write メソッドで `REG_ADDR.ELEPROX_Falling` を指定した時に NCLPROXF 値を使用して書込みます。

有効値の範囲は 0~255 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.ELEPROX_Falling` を指定した時に保持した値
または、SET 処理をした値

FDLPROXF プロパティ

概要

ELEPROX の Filter Delay Count Limit Falling を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.ELEPROX_Falling` を指定した時に保持されます。

Write メソッドで `REG_ADDR.ELEPROX_Falling` を指定した時に FDLPROXF 値を使用して書込みます。

有効値の範囲は 0~255 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.ELEPROX_Falling` を指定した時に保持した値
または、SET 処理をした値

NHDPROXT プロパティ

概要

ELEPROX の Noise Half Delta Touched を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.ELEPROX_Touched` を指定した時に保持されます。

Write メソッドで `REG_ADDR.ELEPROX_Touched` を指定した時に NHDPROXT 値を使用して書込みます。

有効値の範囲は 0~255 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.ELEPROX_Touched` を指定した時に保持した値
または、SET 処理をした値

NCLPROXT プロパティ

概要

ELEPROX の Noise Count Limit Touched を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.ELEPROX_Touched` を指定した時に保持されます。

Write メソッドで `REG_ADDR.ELEPROX_Touched` を指定した時に NCLPROXT 値を使用して書込みます。

有効値の範囲は 0~255 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.ELEPROX_Touched` を指定した時に保持した値
または、SET 処理をした値

FDLPROXT プロパティ

概要

ELEPROX の Filter Delay Count Limit Touched を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.ELEPROX_Touched` を指定した時に保持されます。

Write メソッドで `REG_ADDR.ELEPROX_Touched` を指定した時に FDLPROXT 値を使用して書込みます。

有効値の範囲は 0~255 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.ELEPROX_Touched` を指定した時に保持した値
または、SET 処理をした値

E0TTH プロパティ

概要

ELE0 のタッチ閾値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Threshold` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Threshold` を指定した時に E0TTH 値を使用して書込みます。

有効値の範囲は 0~255 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Threshold` を指定した時に保持した値

または、SET 処理をした値

E0RTH プロパティ

概要

ELE0 のリリース閾値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Threshold` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Threshold` を指定した時に E0RTH 値を使用して書込みます。

有効値の範囲は 0~255 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Threshold` を指定した時に保持した値

または、SET 処理をした値

E1TTH プロパティ

概要

ELE1 のタッチ閾値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Threshold` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Threshold` を指定した時に E1TTH 値を使用して書込みます。

有効値の範囲は 0~255 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Threshold` を指定した時に保持した値

または、SET 処理をした値

E1RTH プロパティ

概要

ELE1 のリリース閾値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Threshold` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Threshold` を指定した時に E1RTH 値を使用して書込みます。

有効値の範囲は 0~255 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Threshold` を指定した時に保持した値

または、SET 処理をした値

E2TTH プロパティ

概要

ELE2 のタッチ閾値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Threshold` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Threshold` を指定した時に E2TTH 値を使用して書込みます。

有効値の範囲は 0~255 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Threshold` を指定した時に保持した値

または、SET 処理をした値

E2RTH プロパティ

概要

ELE2 のリリース閾値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Threshold` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Threshold` を指定した時に E2RTH 値を使用して書込みます。

有効値の範囲は 0~255 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Threshold` を指定した時に保持した値

または、SET 処理をした値

E3TTH プロパティ

概要

ELE3 のタッチ閾値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Threshold` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Threshold` を指定した時に E3TTH 値を使用して書込みます。

有効値の範囲は 0~255 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Threshold` を指定した時に保持した値

または、SET 処理をした値

E3RTH プロパティ

概要

ELE3 のリリース閾値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Threshold` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Threshold` を指定した時に E3RTH 値を使用して書込みます。

有効値の範囲は 0~255 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Threshold` を指定した時に保持した値

または、SET 処理をした値

E4TTH プロパティ

概要

ELE4 のタッチ閾値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Threshold` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Threshold` を指定した時に E4TTH 値を使用して書込みます。

有効値の範囲は 0~255 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Threshold` を指定した時に保持した値

または、SET 処理をした値

E4RTH プロパティ

概要

ELE4 のリリース閾値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Threshold` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Threshold` を指定した時に E4RTH 値を使用して書込みます。

有効値の範囲は 0~255 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Threshold` を指定した時に保持した値

または、SET 処理をした値

E5TTH プロパティ

概要

ELE5 のタッチ閾値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Threshold` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Threshold` を指定した時に E5TTH 値を使用して書込みます。

有効値の範囲は 0~255 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Threshold` を指定した時に保持した値

または、SET 処理をした値

E5RTH プロパティ

概要

ELE5 のリリース閾値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Threshold` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Threshold` を指定した時に E5RTH 値を使用して書込みます。

有効値の範囲は 0~255 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Threshold` を指定した時に保持した値

または、SET 処理をした値

E6TTH プロパティ

概要

ELE6 のタッチ閾値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Threshold` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Threshold` を指定した時に E6TTH 値を使用して書込みます。

有効値の範囲は 0~255 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Threshold` を指定した時に保持した値

または、SET 処理をした値

E6RTH プロパティ

概要

ELE6 のリリース閾値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Threshold` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Threshold` を指定した時に E6RTH 値を使用して書込みます。

有効値の範囲は 0~255 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Threshold` を指定した時に保持した値

または、SET 処理をした値

E7TTH プロパティ

概要

ELE7 のタッチ閾値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Threshold` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Threshold` を指定した時に E7TTH 値を使用して書込みます。

有効値の範囲は 0~255 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Threshold` を指定した時に保持した値

または、SET 処理をした値

E7RTH プロパティ

概要

ELE7 のリリース閾値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Threshold` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Threshold` を指定した時に E7RTH 値を使用して書込みます。

有効値の範囲は 0~255 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Threshold` を指定した時に保持した値

または、SET 処理をした値

E8TTH プロパティ

概要

ELE8 のタッチ閾値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Threshold` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Threshold` を指定した時に E8TTH 値を使用して書込みます。

有効値の範囲は 0~255 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Threshold` を指定した時に保持した値

または、SET 処理をした値

E8RTH プロパティ

概要

ELE8 のリリース閾値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Threshold` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Threshold` を指定した時に E8RTH 値を使用して書込みます。

有効値の範囲は 0~255 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Threshold` を指定した時に保持した値

または、SET 処理をした値

E9TTH プロパティ

概要

ELE9 のタッチ閾値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Threshold` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Threshold` を指定した時に E9TTH 値を使用して書込みます。

有効値の範囲は 0~255 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Threshold` を指定した時に保持した値

または、SET 処理をした値

E9RTH プロパティ

概要

ELE9 のリリース閾値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Threshold` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Threshold` を指定した時に E9RTH 値を使用して書込みます。

有効値の範囲は 0~255 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Threshold` を指定した時に保持した値

または、SET 処理をした値

E10TTH プロパティ

概要

ELE10 のタッチ閾値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Threshold` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Threshold` を指定した時に E10TTH 値を使用して書込みます。

有効値の範囲は 0~255 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Threshold` を指定した時に保持した値

または、SET 処理をした値

E10RTH プロパティ

概要

ELE10 のリリース閾値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Threshold` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Threshold` を指定した時に E10RTH 値を使用して書込みます。

有効値の範囲は 0~255 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Threshold` を指定した時に保持した値

または、SET 処理をした値

E11TTH プロパティ

概要

ELE11 のタッチ閾値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Threshold` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Threshold` を指定した時に E11TTH 値を使用して書込みます。

有効値の範囲は 0~255 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Threshold` を指定した時に保持した値

または、SET 処理をした値

E11RTH プロパティ

概要

ELE11 のリリース閾値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Threshold` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Threshold` を指定した時に E11RTH 値を使用して書込みます。

有効値の範囲は 0~255 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Threshold` を指定した時に保持した値

または、SET 処理をした値

E12TTH プロパティ

概要

ELEPROX のタッチ閾値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Threshold` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Threshold` を指定した時に ELEPROX 値を使用して書込みます。

有効値の範囲は 0~255 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Threshold` を指定した時に保持した値

または、SET 処理をした値

E12RTH プロパティ

概要

ELEPROX のリリース閾値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Threshold` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Threshold` を指定した時に ELEPROX 値を使用して書込みます。

有効値の範囲は 0~255 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Threshold` を指定した時に保持した値

または、SET 処理をした値

DR プロパティ

概要

リリース用のデバウンス数を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Debounce` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Debounce` を指定した時に DR 値を使用して書込みます。

値の範囲は 0~7 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Debounce` を指定した時に保持した値

または、SET 処理をした値

DT プロパティ

概要

タッチ用のデバウンス数を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Debounce` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Debounce` を指定した時に DT 値を使用して書込みます。

値の範囲は 0~7 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Debounce` を指定した時に保持した値

または、SET 処理をした値

FFI プロパティ

概要

First Filter Iterations を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.AF_Configuration_1` を指定した時に保持されます。

Write メソッドで `REG_ADDR.AF_Configuration_1` を指定した時に FFI 値を使用して書込みます。

設定値 (0~3)

「00」0: Sets samples taken to 6 (Default)

「01」1: Sets samples taken to 10

「10」2: Sets samples taken to 18

「11」3: Sets samples taken to 34

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.AF_Configuration_1` を指定した時に保持した値

または、SET 処理をした値

CDC プロパティ

概要

Charge Discharge Current を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.AF_Configuration_1` を指定した時に保持されます。

Write メソッドで `REG_ADDR.AF_Configuration_1` を指定した時に CDC 値を使用して書込み
す。

設定値 (0~63) 1 μ A 単位

「000000」0: 充電無効

「000001」1: 電流設定 1 μ A

「010000」16: 電流設定 16 μ A(Default)

「111111」63: 電流設定 63 μ A

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.AF_Configuration_1` を指定した時に保持した値
または、SET 処理をした値

CDT プロパティ

概要

Charge Discharge Time を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.AF_Configuration_2` を指定した時に保持されます。

Write メソッドで `REG_ADDR.AF_Configuration_2` を指定した時に CDT 値を使用して書込み
す。

設定値 (0~7) $0.5 (2^{n-1}) \mu\text{s}$ 単位

「000」0: 無効

「001」1: $0.5 \mu\text{s}$ (Default)

「010」2: $1 \mu\text{s}$

 ~: $0.5 (2^{n-1}) \mu\text{s}$

「111」7: $32 \mu\text{s}$

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.AF_Configuration_2` を指定した時に保持した値
または、SET 処理をした値

SFI プロパティ

概要

Second Filter Iterations を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.AF_Configuration_2` を指定した時に保持されます。

Write メソッドで `REG_ADDR.AF_Configuration_2` を指定した時に SFI 値を使用して書込みます。

設定値 (0~3)

「00」0: Number of samples is set to 4 (Default)

「01」1: Number of samples is set to 6

「10」2: Number of samples is set to 10

「11」3: Number of samples is set to 18

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.AF_Configuration_2` を指定した時に保持した値

または、SET 処理をした値

ESI プロパティ

概要

Electrode Sample Interval を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.AF_Configuration_2` を指定した時に保持されます。

Write メソッドで `REG_ADDR.AF_Configuration_2` を指定した時に ESI 値を使用して書込みます。

設定値 (0~7) 2^n ms 単位

「000」0:1ms

「001」1:2ms

「100」4:16ms (default)

 ~: 2^n ms

「111」7:128ms

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.AF_Configuration_2` を指定した時に保持した値

または、SET 処理をした値

CL プロパティ

概要

Calibration Lock bit を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Electrode_Configuration` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Electrode_Configuration` を指定した時に CL 値を使用して書込み
す。

設定値 (0~3)

「00」0: Baseline tracking enable (Default).

「01」1: Calibration lock. Baseline tracking disabled

「10」2: Baseline tracking and initialize enable.

「11」3: Baseline tracking and initialize enable.

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Electrode_Configuration` を指定した時に保持した値
または、SET 処理をした値

ELEPROX プロパティ

概要

ELEPROX ピンの組み合わせを取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Electrode_Configuration` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Electrode_Configuration` を指定した時に ELEPROX 値を使用して書込ます。

設定値 (0~3)

「00」0: Proximity detection disabled(Default)

「01」1: ELE0~1 を有効

「10」2: ELE0~3 を有効

「11」3: ELE0~11 を有効

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Electrode_Configuration` を指定した時に保持した値

または、SET 処理をした値

ELE プロパティ

概要

各ピンの電極タッチ検出の有効／無効を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Electrode_Configuration` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Electrode_Configuration` を指定した時に ELE 値を使用して書込みます。

設定値 (0~15)

「0000」0: Electrode touch detection disabled (Default)

「0001」1: ELE0 を有効

「0010」2: ELE0~1 を有効

~

「1011」11: ELE0~10 を有効

「11xx」12~15: ELE0~11 を有効

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Electrode_Configuration` を指定した時に保持した値
または、SET 処理をした値

CDC0 プロパティ

概要

ELE0 の Electrode Current を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Electrode_Current` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Electrode_Current` を指定した時に CDC0 値を使用して書込み
す。

設定値 (0~63) 1 μ A 単位

「000000」0: Global CDC を使用(Default)

「000001」1: 電流設定 1 μ A

「010000」16: 電流設定 16 μ A

「111111」63: 電流設定 63 μ A

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Electrode_Current` を指定した時に保持した値
または、SET 処理をした値

CDC1 プロパティ

概要

ELE1 の Electrode Current を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Electrode_Current` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Electrode_Current` を指定した時に CDC1 値を使用して書込み
す。

設定値 (0~63) 1 μ A 単位

「000000」0: Global CDC を使用(Default)

「000001」1: 電流設定 1 μ A

「010000」16: 電流設定 16 μ A

「111111」63: 電流設定 63 μ A

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Electrode_Current` を指定した時に保持した値
または、SET 処理をした値

CDC2 プロパティ

概要

ELE2 の Electrode Current を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Electrode_Current` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Electrode_Current` を指定した時に CDC2 値を使用して書込み
す。

設定値 (0~63) 1 μ A 単位

「000000」0: Global CDC を使用(Default)

「000001」1: 電流設定 1 μ A

「010000」16: 電流設定 16 μ A

「111111」63: 電流設定 63 μ A

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Electrode_Current` を指定した時に保持した値
または、SET 処理をした値

CDC3 プロパティ

概要

ELE3 の Electrode Current を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Electrode_Current` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Electrode_Current` を指定した時に CDC3 値を使用して書込み
す。

設定値 (0~63) 1 μ A 単位

「000000」0: Global CDC を使用(Default)

「000001」1: 電流設定 1 μ A

「010000」16: 電流設定 16 μ A

「111111」63: 電流設定 63 μ A

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Electrode_Current` を指定した時に保持した値
または、SET 処理をした値

CDC4 プロパティ

概要

ELE4 の Electrode Current を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Electrode_Current` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Electrode_Current` を指定した時に CDC4 値を使用して書込み
す。

設定値 (0~63) 1 μ A 単位

「000000」0: Global CDC を使用(Default)

「000001」1: 電流設定 1 μ A

「010000」16: 電流設定 16 μ A

「111111」63: 電流設定 63 μ A

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Electrode_Current` を指定した時に保持した値
または、SET 処理をした値

CDC5 プロパティ

概要

ELE5 の Electrode Current を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Electrode_Current` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Electrode_Current` を指定した時に CDC5 値を使用して書込み
す。

設定値 (0~63) 1 μ A 単位

「000000」0: Global CDC を使用(Default)

「000001」1: 電流設定 1 μ A

「010000」16: 電流設定 16 μ A

「111111」63: 電流設定 63 μ A

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Electrode_Current` を指定した時に保持した値
または、SET 処理をした値

CDC6 プロパティ

概要

ELE6 の Electrode Current を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Electrode_Current` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Electrode_Current` を指定した時に CDC6 値を使用して書込み
す。

設定値 (0~63) 1 μ A 単位

「000000」0: Global CDC を使用(Default)

「000001」1: 電流設定 1 μ A

「010000」16: 電流設定 16 μ A

「111111」63: 電流設定 63 μ A

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Electrode_Current` を指定した時に保持した値
または、SET 処理をした値

CDC7 プロパティ

概要

ELE7 の Electrode Current を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Electrode_Current` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Electrode_Current` を指定した時に CDC7 値を使用して書込み
す。

設定値 (0~63) 1 μ A 単位

「000000」0: Global CDC を使用(Default)

「000001」1: 電流設定 1 μ A

「010000」16: 電流設定 16 μ A

「111111」63: 電流設定 63 μ A

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Electrode_Current` を指定した時に保持した値
または、SET 処理をした値

CDC8 プロパティ

概要

ELE8 の Electrode Current を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Electrode_Current` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Electrode_Current` を指定した時に CDC8 値を使用して書込み
す。

設定値 (0~63) 1 μ A 単位

「000000」0: Global CDC を使用(Default)

「000001」1: 電流設定 1 μ A

「010000」16: 電流設定 16 μ A

「111111」63: 電流設定 63 μ A

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Electrode_Current` を指定した時に保持した値
または、SET 処理をした値

CDC9 プロパティ

概要

ELE9 の Electrode Current を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Electrode_Current` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Electrode_Current` を指定した時に CDC9 値を使用して書込みます。

設定値 (0~63) 1 μ A 単位

「000000」0: Global CDC を使用(Default)

「000001」1: 電流設定 1 μ A

「010000」16: 電流設定 16 μ A

「111111」63: 電流設定 63 μ A

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Electrode_Current` を指定した時に保持した値
または、SET 処理をした値

CDC10 プロパティ

概要

ELE10 の Electrode Current を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Electrode_Current` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Electrode_Current` を指定した時に CDC10 値を使用して書込み
す。

設定値 (0~63) 1 μ A 単位

「000000」0: Global CDC を使用(Default)

「000001」1: 電流設定 1 μ A

「010000」16: 電流設定 16 μ A

「111111」63: 電流設定 63 μ A

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Electrode_Current` を指定した時に保持した値
または、SET 処理をした値

CDC11 プロパティ

概要

ELE11 の Electrode Current を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Electrode_Current` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Electrode_Current` を指定した時に CDC11 値を使用して書込ます。

設定値 (0~63) 1 μ A 単位

「000000」0: Global CDC を使用(Default)

「000001」1: 電流設定 1 μ A

「010000」16: 電流設定 16 μ A

「111111」63: 電流設定 63 μ A

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Electrode_Current` を指定した時に保持した値
または、SET 処理をした値

CDC12 プロパティ

概要

ELEPROX の Electrode Current を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Electrode_Current` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Electrode_Current` を指定した時に ELEPROX 値を使用して書込みます。

設定値 (0~63) 1 μ A 単位

「000000」0: Global CDC を使用(Default)

「000001」1: 電流設定 1 μ A

「010000」16: 電流設定 16 μ A

「111111」63: 電流設定 63 μ A

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Electrode_Current` を指定した時に保持した値
または、SET 処理をした値

CDT0 プロパティ

概要

ELE0 の Charge Time を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Charge_Time` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Charge_Time` を指定した時に CDT0 値を使用して書込みます。

設定値(0~7) $0.5 (2^{n-1}) \mu\text{s}$ 単位

「000」0: Global CDC を使用(Default)

「001」1: $0.5 \mu\text{s}$

「010」2: $1 \mu\text{s}$

 ~: $0.5 (2^{n-1}) \mu\text{s}$

「111」7: $32 \mu\text{s}$

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Charge_Time` を指定した時に保持した値

または、SET 処理をした値

CDT1 プロパティ

概要

ELE1 の Charge Time を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Charge_Time` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Charge_Time` を指定した時に CDT1 値を使用して書込みます。

設定値(0~7) $0.5 (2^{n-1}) \mu\text{s}$ 単位

「000」0: Global CDC を使用(Default)

「001」1: $0.5 \mu\text{s}$

「010」2: $1 \mu\text{s}$

 ~: $0.5 (2^{n-1}) \mu\text{s}$

「111」7: $32 \mu\text{s}$

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Charge_Time` を指定した時に保持した値

または、SET 処理をした値

CDT2 プロパティ

概要

ELE2 の Charge Time を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Charge_Time` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Charge_Time` を指定した時に CDT2 値を使用して書込みます。

設定値(0~7) $0.5 (2^{n-1}) \mu\text{s}$ 単位

「000」0: Global CDC を使用(Default)

「001」1: $0.5 \mu\text{s}$

「010」2: $1 \mu\text{s}$

 ~: $0.5 (2^{n-1}) \mu\text{s}$

「111」7: $32 \mu\text{s}$

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Charge_Time` を指定した時に保持した値

または、SET 処理をした値

CDT3 プロパティ

概要

ELE3 の Charge Time を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Charge_Time` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Charge_Time` を指定した時に CDT3 値を使用して書込みます。

設定値(0~7) $0.5 (2^{n-1}) \mu\text{s}$ 単位

「000」0: Global CDC を使用(Default)

「001」1: $0.5 \mu\text{s}$

「010」2: $1 \mu\text{s}$

 ~: $0.5 (2^{n-1}) \mu\text{s}$

「111」7: $32 \mu\text{s}$

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Charge_Time` を指定した時に保持した値

または、SET 処理をした値

CDT4 プロパティ

概要

ELE4 の Charge Time を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Charge_Time` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Charge_Time` を指定した時に CDT4 値を使用して書込みます。

設定値(0~7) $0.5 (2^{n-1}) \mu\text{s}$ 単位

「000」0: Global CDC を使用(Default)

「001」1: $0.5 \mu\text{s}$

「010」2: $1 \mu\text{s}$

 ~: $0.5 (2^{n-1}) \mu\text{s}$

「111」7: $32 \mu\text{s}$

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Charge_Time` を指定した時に保持した値

または、SET 処理をした値

CDT5 プロパティ

概要

ELE5 の Charge Time を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Charge_Time` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Charge_Time` を指定した時に CDT5 値を使用して書込みます。

設定値(0~7) $0.5 (2^{n-1}) \mu\text{s}$ 単位

「000」0: Global CDC を使用(Default)

「001」1: $0.5 \mu\text{s}$

「010」2: $1 \mu\text{s}$

 ~: $0.5 (2^{n-1}) \mu\text{s}$

「111」7: $32 \mu\text{s}$

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Charge_Time` を指定した時に保持した値

または、SET 処理をした値

CDT6 プロパティ

概要

ELE6 の Charge Time を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Charge_Time` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Charge_Time` を指定した時に CDT6 値を使用して書込みます。

設定値(0~7) $0.5 (2^{n-1}) \mu\text{s}$ 単位

「000」0: Global CDC を使用(Default)

「001」1: $0.5 \mu\text{s}$

「010」2: $1 \mu\text{s}$

 ~: $0.5 (2^{n-1}) \mu\text{s}$

「111」7: $32 \mu\text{s}$

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Charge_Time` を指定した時に保持した値

または、SET 処理をした値

CDT7 プロパティ

概要

ELE7 の Charge Time を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Charge_Time` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Charge_Time` を指定した時に CDT7 値を使用して書込みます。

設定値(0~7) $0.5 (2^{n-1}) \mu\text{s}$ 単位

「000」0: Global CDC を使用(Default)

「001」1: $0.5 \mu\text{s}$

「010」2: $1 \mu\text{s}$

 ~: $0.5 (2^{n-1}) \mu\text{s}$

「111」7: $32 \mu\text{s}$

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Charge_Time` を指定した時に保持した値

または、SET 処理をした値

CDT8 プロパティ

概要

ELE8 の Charge Time を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Charge_Time` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Charge_Time` を指定した時に CDT8 値を使用して書込みます。

設定値(0~7) $0.5 (2^{n-1}) \mu\text{s}$ 単位

「000」0: Global CDC を使用(Default)

「001」1: $0.5 \mu\text{s}$

「010」2: $1 \mu\text{s}$

 ~: $0.5 (2^{n-1}) \mu\text{s}$

「111」7: $32 \mu\text{s}$

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Charge_Time` を指定した時に保持した値

または、SET 処理をした値

CDT9 プロパティ

概要

ELE9 の Charge Time を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Charge_Time` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Charge_Time` を指定した時に CDT9 値を使用して書込みます。

設定値(0~7) $0.5 (2^{n-1}) \mu\text{s}$ 単位

「000」0: Global CDC を使用(Default)

「001」1: $0.5 \mu\text{s}$

「010」2: $1 \mu\text{s}$

 ~: $0.5 (2^{n-1}) \mu\text{s}$

「111」7: $32 \mu\text{s}$

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Charge_Time` を指定した時に保持した値

または、SET 処理をした値

CDT10 プロパティ

概要

ELE10 の Charge Time を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Charge_Time` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Charge_Time` を指定した時に CDT10 値を使用して書込みます。

設定値(0~7) $0.5 (2^{n-1}) \mu\text{s}$ 単位

「000」0: Global CDC を使用(Default)

「001」1: $0.5 \mu\text{s}$

「010」2: $1 \mu\text{s}$

 ~: $0.5 (2^{n-1}) \mu\text{s}$

「111」7: $32 \mu\text{s}$

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Charge_Time` を指定した時に保持した値

または、SET 処理をした値

CDT11 プロパティ

概要

ELE11 の Charge Time を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Charge_Time` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Charge_Time` を指定した時に CDT11 値を使用して書込みます。

設定値 (0~7) $0.5 (2^{n-1}) \mu\text{s}$ 単位

「000」0: Global CDC を使用(Default)

「001」1: $0.5 \mu\text{s}$

「010」2: $1 \mu\text{s}$

 ~: $0.5 (2^{n-1}) \mu\text{s}$

「111」7: $32 \mu\text{s}$

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Charge_Time` を指定した時に保持した値

または、SET 処理をした値

CDT12 プロパティ

概要

ELEPROX の Charge Time を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Charge_Time` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Charge_Time` を指定した時に ELEPROX 値を使用して書込みます。

設定値 (0~7) $0.5 (2^{n-1}) \mu\text{s}$ 単位

「000」0: Global CDC を使用(Default)

「001」1: $0.5 \mu\text{s}$

「010」2: $1 \mu\text{s}$

~: $0.5 (2^{n-1}) \mu\text{s}$

「111」7: $32 \mu\text{s}$

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Charge_Time` を指定した時に保持した値

または、SET 処理をした値

CTL0 プロパティ

概要

GPIO Control Register 0 を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.GPIO` を指定した時に保持されます。

Write メソッドで `REG_ADDR.GPIO` を指定した時に CTL0 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.GPIO` を指定した時に保持した値
または、SET 処理をした値

CTL1 プロパティ

概要

GPIO Control Register 1 を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.GPIO` を指定した時に保持されます。

Write メソッドで `REG_ADDR.GPIO` を指定した時に CTL1 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.GPIO` を指定した時に保持した値
または、SET 処理をした値

DAT プロパティ

概要

GPIO Data Register を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.GPIO` を指定した時に保持されます。

Write メソッドで `REG_ADDR.GPIO` を指定した時に DAT 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.GPIO` を指定した時に保持した値
または、SET 処理をした値

DIR プロパティ

概要

GPIO Direction Control Register を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.GPIO` を指定した時に保持されます。

Write メソッドで `REG_ADDR.GPIO` を指定した時に DIR 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.GPIO` を指定した時に保持した値
または、SET 処理をした値

EN プロパティ

概要

GPIO Enable Register を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.GPIO` を指定した時に保持されます。

Write メソッドで `REG_ADDR.GPIO` を指定した時に EN 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.GPIO` を指定した時に保持した値
または、SET 処理をした値

SET プロパティ

概要

GPIO Data Set Register を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.GPIO` を指定した時に保持されます。

Write メソッドで `REG_ADDR.GPIO` を指定した時に SET 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.GPIO` を指定した時に保持した値
または、SET 処理をした値

CLR プロパティ

概要

GPIO Data Clear Register を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.GPIO` を指定した時に保持されます。

Write メソッドで `REG_ADDR.GPIO` を指定した時に CLR 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.GPIO` を指定した時に保持した値
または、SET 処理をした値

TOG プロパティ

概要

GPIO Data Toggle Register を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.GPIO` を指定した時に保持されます。

Write メソッドで `REG_ADDR.GPIO` を指定した時に TOG 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.GPIO` を指定した時に保持した値
または、SET 処理をした値

AFES プロパティ

概要

AFE Samples during binary search を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.AUTO_CONFIG` を指定した時に保持されます。

Write メソッドで `REG_ADDR.AUTO_CONFIG` を指定した時に AFES 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.AUTO_CONFIG` を指定した時に保持した値
または、SET 処理をした値

RETRY プロパティ

概要

PR121 may retry AUTO-CONFIG or AUTO-RECONFIG after failed を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.AUTO_CONFIG` を指定した時に保持されます。

Write メソッドで `REG_ADDR.AUTO_CONFIG` を指定した時に RETRY 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.AUTO_CONFIG` を指定した時に保持した値
または、SET 処理をした値

BVA プロパティ

概要

Baseline Value Adjust Control を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.AUTO_CONFIG` を指定した時に保持されます。

Write メソッドで `REG_ADDR.AUTO_CONFIG` を指定した時に BVA 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.AUTO_CONFIG` を指定した時に保持した値
または、SET 処理をした値

ARE プロパティ

概要

AUTO-RECONFIG の有効／無効を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.AUTO_CONFIG` を指定した時に保持されます。

Write メソッドで `REG_ADDR.AUTO_CONFIG` を指定した時に ARE 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.AUTO_CONFIG` を指定した時に保持した値
または、SET 処理をした値

ACE プロパティ

概要

AUTO-CONFIG の有効／無効を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.AUTO_CONFIG` を指定した時に保持されます。

Write メソッドで `REG_ADDR.AUTO_CONFIG` を指定した時に ACE 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.AUTO_CONFIG` を指定した時に保持した値
または、SET 処理をした値

SCTS プロパティ

概要

Skip Charge Time Search を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.AUTO_CONFIG` を指定した時に保持されます。

Write メソッドで `REG_ADDR.AUTO_CONFIG` を指定した時に SCTS 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.AUTO_CONFIG` を指定した時に保持した値
または、SET 処理をした値

OORIE プロパティ

概要

Out of Range Interrupt Enable を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.AUTO_CONFIG` を指定した時に保持されます。

Write メソッドで `REG_ADDR.AUTO_CONFIG` を指定した時に OORIE 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.AUTO_CONFIG` を指定した時に保持した値
または、SET 処理をした値

ARFIE プロパティ

概要

AUTO-RECONFIG Fail Interrupt Enable を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.AUTO_CONFIG` を指定した時に保持されます。

Write メソッドで `REG_ADDR.AUTO_CONFIG` を指定した時に ARFIE 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.AUTO_CONFIG` を指定した時に保持した値
または、SET 処理をした値

ACFIE プロパティ

概要

AUTO-CONFIG Fail Interrupt Enable を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.AUTO_CONFIG` を指定した時に保持されます。

Write メソッドで `REG_ADDR.AUTO_CONFIG` を指定した時に ACFIE 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.AUTO_CONFIG` を指定した時に保持した値
または、SET 処理をした値

USL プロパティ

概要

AUTO-CONFIG USL Register を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.AUTO_CONFIG` を指定した時に保持されます。

Write メソッドで `REG_ADDR.AUTO_CONFIG` を指定した時に USL 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.AUTO_CONFIG` を指定した時に保持した値
または、SET 処理をした値

LSL プロパティ

概要

AUTO-CONFIG LSL Register を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.AUTO_CONFIG` を指定した時に保持されます。

Write メソッドで `REG_ADDR.AUTO_CONFIG` を指定した時に LSL 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.AUTO_CONFIG` を指定した時に保持した値
または、SET 処理をした値

TL プロパティ

概要

Target Level を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.AUTO_CONFIG` を指定した時に保持されます。

Write メソッドで `REG_ADDR.AUTO_CONFIG` を指定した時に TL 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.AUTO_CONFIG` を指定した時に保持した値
または、SET 処理をした値

SRST プロパティ

概要

電源オン後の状態にリセットします。

処理内容

Read メソッドで `REG_ADDR.Soft_Reset` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Soft_Reset` を指定した時に SRST 値を使用して書込みます。

設定値: 0x63 電源オン後の状態にリセット

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Soft_Reset` を指定した時に保持した値
または、SET 処理をした値

ReadBuff プロパティ

概要

Read メソッド時の読込データをセットします。

処理内容

Read メソッド時に保持されます。

SET 処理

なし

GET 処理

Read メソッド時に保持した値

Init メソッド

概要

I2C バスコントローラに指定された接続設定を使って非同期的に接続します。

引数

| 引数名 | 型 | 説明 |
|---------|------|-----------|
| Address | byte | I2C のアドレス |

戻り値

なし

Read メソッド

概要

最新の変換結果を読みます。

引数

| 引数名 | 型 | 説明 |
|-----------|------|--|
| RegSelect | byte | REG_ADDR_Touch_Status (0x00) ~ REG_ADDR_Soft_Reset (0x80) |

戻り値

Status と同じ

1: 正常

1 以外: 異常

読込処理

| REG_ADDR 値 | 読込プロパティ |
|---------------------|--|
| Touch_Status | E0TS E1TS E2TS E3TS E4TS E5TS E6TS E7TS E8TS E9TS E10TS E11TS E12TS OVCF |
| Out_Of_Range_Status | E0OOR E1OOR E2OOR E3OOR E4OOR E5OOR E6OOR E7OOR E8OOR E9OOR E10OOR E11OOR E12OOR ACFF ARFF |

| REG_ADDR 値 | 読込プロパティ |
|----------------|---------|
| Electrode_Data | E0FD |
| | E1FD |
| | E2FD |
| | E3FD |
| | E4FD |
| | E5FD |
| | E6FD |
| | E7FD |
| | E8FD |
| | E9FD |
| | E10FD |
| | E11FD |
| | E12FD |
| Baseline_Value | E0BV |
| | E1BV |
| | E2BV |
| | E3BV |
| | E4BV |
| | E5BV |
| | E6BV |
| | E7BV |
| | E8BV |
| | E9BV |
| | E10BV |
| | E11BV |
| | E12BV |

| REG_ADDR 値 | 読込プロパティ |
|-----------------|--|
| Rising | MHDR NHDR NCLR FDLR |
| Falling | MHDF NHDF NCLF FDLF |
| Touched | NHDT NCLT FDLT |
| ELEPROX_Rising | MHDPROXR NHDPROXR NCLPROXR FDLPROXR |
| ELEPROX_Falling | MHDPROXF NHDPROXF NCLPROXF FDLPROXF |
| ELEPROX_Touched | NHDPROXT NCLPROXT FDLPROXT |

| REG_ADDR 値 | 読込プロパティ |
|------------|---------|
| Threshold | E0TTH |
| | E0RTH |
| | E1TTH |
| | E1RTH |
| | E2TTH |
| | E2RTH |
| | E3TTH |
| | E3RTH |
| | E4TTH |
| | E4RTH |
| | E5TTH |
| | E5RTH |
| | E6TTH |
| | E6RTH |
| | E7TTH |
| | E7RTH |
| | E8TTH |
| | E8RTH |
| | E9TTH |
| | E9RTH |
| | E10TTH |
| | E10RTH |
| | E11TTH |
| | E11RTH |
| E12TTH | |
| E12RTH | |

| REG_ADDR 値 | 読込プロパティ |
|-------------------------|---|
| Debounce | DR DT |
| AF_Configuration_1 | FFI CDC |
| AF_Configuration_2 | CDT SFI ESI |
| Electrode_Configuration | CL ELEPROX ELE |
| Electrode_Current | CDC0 CDC1 CDC2 CDC3 CDC4 CDC5 CDC6 CDC7 CDC8 CDC9 CDC10 CDC11 CDC12 |

| REG_ADDR 値 | 読込プロパティ |
|-------------|---|
| Charge_Time | CDT0 CDT1 CDT2 CDT3 CDT4 CDT5 CDT6 CDT7 CDT8 CDT9 CDT10 CDT11 CDT12 |
| GPIO | CTL0 CTL1 DAT DIR EN SET CLR TOG |

| REG_ADDR 値 | 読込プロパティ |
|-------------|---|
| AUTO_CONFIG | AFES RETRY BVA ARE ACE SCTS OORIE ARFIE ACFIE USL LSL TL |
| Soft_Reset | SRST |

Write メソッド

概要

引数 RegSelect に関するプロパティを編集し書き込みます。

引数

| 引数名 | 型 | 説明 |
|-----------|------|--|
| RegSelect | byte | REG_ADDR_Touch_Status (0x00) ~ REG_ADDR_Soft_Reset (0x80) |

戻り値

Status

1: 正常

1 以外: 異常

書込処理

| REG_ADDR 値 | 書込プロパティ |
|-----------------|--|
| Rising | MHDR NHDR NCLR FDLR |
| Falling | MHDF NHDF NCLF FDLF |
| Touched | NHDT NCLT FDLT |
| ELEPROX_Rising | MHDPROXR NHDPROXR NCLPROXR FDLPROXR |
| ELEPROX_Falling | MHDPROXF NHDPROXF NCLPROXF FDLPROXF |
| ELEPROX_Touched | NHDPROXT NCLPROXT FDLPROXT |

| REG_ADDR 値 | 書込プロパティ |
|------------|---------|
| Threshold | E0TTH |
| | E0RTH |
| | E1TTH |
| | E1RTH |
| | E2TTH |
| | E2RTH |
| | E3TTH |
| | E3RTH |
| | E4TTH |
| | E4RTH |
| | E5TTH |
| | E5RTH |
| | E6TTH |
| | E6RTH |
| | E7TTH |
| | E7RTH |
| | E8TTH |
| | E8RTH |
| | E9TTH |
| | E9RTH |
| | E10TTH |
| | E10RTH |
| | E11TTH |
| | E11RTH |
| E12TTH | |
| E12RTH | |

| REG_ADDR 値 | 書込プロパティ |
|-------------------------|---|
| Debounce | DR DT |
| AF_Configuration_1 | FFI CDC |
| AF_Configuration_2 | CDT SFI ESI |
| Electrode_Configuration | CL ELEPROX ELE |
| Electrode_Current | CDC0 CDC1 CDC2 CDC3 CDC4 CDC5 CDC6 CDC7 CDC8 CDC9 CDC10 CDC11 CDC12 |

| REG_ADDR 値 | 書込プロパティ |
|-------------|---|
| Charge_Time | CDT0 CDT1 CDT2 CDT3 CDT4 CDT5 CDT6 CDT7 CDT8 CDT9 CDT10 CDT11 CDT12 |
| GPIO | CTL0 CTL1 DAT DIR EN SET CLR TOG |

| REG_ADDR 値 | 書込プロパティ |
|-------------|---|
| AUTO_CONFIG | AFES RETRY BVA ARE ACE SCTS OORIE ARFIE ACFIE USL LSL TL |
| Soft_Reset | SRST |

サンプルプログラム

```
Imports I2CLibrary
Public NotInheritable Class MPR121Page
    Inherits Page

    Private Const MPR121_Address As Byte = &H5A
    Private MPR121 As MPR121 = New MPR121()
    Private WithEvents RefreshTimer As DispatcherTimer = New DispatcherTimer()

    Private Const TOU_THRESH As Byte = &H6
    Private Const REL_THRESH As Byte = &HA

    Public Sub New()
        InitializeComponent()
        Call Init()
    End Sub

    Private Async Sub Init()
        Await MPR121.Init(MPR121_Address)

        ' Rising書込
        MPR121.MHDR = &H1
        MPR121.NHDR = &H1
        MPR121.NCLR = &H0
        MPR121.FDLR = &H0
        MPR121.Write(MPR121.REG_ADDR.Rising)

        ' Falling書込
        MPR121.MHDF = &H1
        MPR121.NHDF = &H1
        MPR121.NCLF = &HFF
        MPR121.FDLF = &H2
        MPR121.Write(MPR121.REG_ADDR.Falling)

        MPR121.E0TTH = TOU_THRESH
        MPR121.E0RTH = REL_THRESH
        MPR121.E1TTH = TOU_THRESH
        MPR121.E1RTH = REL_THRESH
        MPR121.E2TTH = TOU_THRESH
        MPR121.E2RTH = REL_THRESH
        MPR121.E3TTH = TOU_THRESH
        MPR121.E3RTH = REL_THRESH
        MPR121.E4TTH = TOU_THRESH
        MPR121.E4RTH = REL_THRESH
        MPR121.E5TTH = TOU_THRESH
        MPR121.E5RTH = REL_THRESH
        MPR121.E6TTH = TOU_THRESH
        MPR121.E6RTH = REL_THRESH
        MPR121.E7TTH = TOU_THRESH
    End Sub
End Class
```

```

MPR121.E7RTH = REL_THRESH
MPR121.E8TTH = TOU_THRESH
MPR121.E8RTH = REL_THRESH
MPR121.E9TTH = TOU_THRESH
MPR121.E9RTH = REL_THRESH
MPR121.E10TTH = TOU_THRESH
MPR121.E10RTH = REL_THRESH
MPR121.E11TTH = TOU_THRESH
MPR121.E11RTH = REL_THRESH
MPR121.E12TTH = TOU_THRESH
MPR121.E12RTH = REL_THRESH
MPR121.Write(MPR121.REG_ADDR.Threshold)

```

```

MPR121.CDT = 0
MPR121.SFI = 0
MPR121.ESI = &H4
MPR121.Write(MPR121.REG_ADDR.AF_Configuration_2)

```

```

MPR121.CL = 0
MPR121.ELEPROX = 0
MPR121.ELE = &HC
MPR121.Write(MPR121.REG_ADDR.Electrode_Configuration)

```

End Sub

```

Private Sub btnStart_Click(sender As Object, e As RoutedEventArgs)
    RefreshTimer.Interval = TimeSpan.FromMilliseconds(100)
    RefreshTimer.Start()

```

End Sub

```

Private Sub Measurement(sender As Object, e As Object) Handles RefreshTimer.Tick
    RefreshTimer.Stop()

```

```

MPR121.Read(MPR121.REG_ADDR.Touch_Status)
If (MPR121.Status = 1) Then
    If (MPR121.E0TS) Then
        Debug.WriteLine("E0TS : detected")
    End If
    If (MPR121.E1TS) Then
        Debug.WriteLine("E1TS : detected")
    End If
    If (MPR121.E2TS) Then
        Debug.WriteLine("E2TS : detected")
    End If
    If (MPR121.E3TS) Then
        Debug.WriteLine("E3TS : detected")
    End If
End If

```

```

RefreshTimer.Start()

```

End Sub

```

Private Sub btnStop_Click(sender As Object, e As RoutedEventArgs)
    RefreshTimer.Stop()

```

End Sub

```
Private Sub btnBack_Click(sender As Object, e As RoutedEventArgs)
    RefreshTimer.Stop()
    MPR121.close()
    App.rootFrame.Navigate(GetType(MenuPage))
End Sub
```

```
End Class
```

3-5.12bit D/A コンバータ MCP4275

変更履歴

2016年9月 15日 第一版 新規作成

1. 概要

12-BitD/A コンバータ(MCP4725)とI2C 接続するための機能を提供します。

2. 機能一覧

プロパティ

| 機能名 | 機能説明 |
|------------------------|--|
| RDYBSY | RDY/ BSY |
| POR | power-on-reset |
| PowerDownSelect | 0: Normal Mode 1: 1 k Ω resistor to ground 2: 100 k Ω resistor to ground 3: 500 k Ω resistor to ground |
| register_Data | DAC Input Data (12 bits) |
| EEPROM_PowerDownSelect | 0: Normal Mode 1: 1 k Ω resistor to ground 2: 100 k Ω resistor to ground 3: 500 k Ω resistor to ground |
| EEPROM_Data | DAC Input Data (12 bits) |
| ReadBuff | 読込生データ |

メソッド

| 機能名 | 機能説明 |
|-------|---------------------------------------|
| Init | I2C バスコントローラに指定された接続設定を使って非同期的に接続します。 |
| Read | 指定レジスタ読込 |
| Write | 指定レジスタ書込 |

定数

| 機能名 | 機能説明 | 定数值 |
|-----------------|----------------------|------|
| power_down_mode | Normal | 0x00 |
| | ground1k | 0x01 |
| | ground100k | 0x02 |
| | ground500k | 0x03 |
| CommandType | FastMode | 0x00 |
| | DACRegister | 0x02 |
| | DACRegisterAndEEPROM | 0x03 |

RDYBSY プロパティ

概要

RDY/BSY の値を取得または設定します。

処理内容

Read メソッドを指定した時に保持されます。

SET 処理

なし

GET 処理

Read メソッド時に保持した値

POR プロパティ

概要

POR の値を取得または設定します。

処理内容

Read メソッドを指定した時に保持されます。

SET 処理

なし

GET 処理

Read メソッド時に保持した値

register_Data プロパティ

概要

DAC Input Data (12 bits)の値を取得または設定します。

処理内容

Read メソッド時に保持されます。

Write メソッドで DAC_Input_Data 値を使用して書込みます。

`CommandType.FastMode`: DAC レジスタを変更する。EEPROM は影響を受けません

`CommandType.DACRegister`: コンフィギュレーションビットとデータコードを DAC レジスタにロードする

`CommandType.DACRegisterAndEEPROM`: コンフィギュレーションビットとデータコードを DAC レジスタにロードし、EEPROM に書き込む

SET 処理

value 値をセットする。

GET 処理

Read メソッド時に保持した値

または、SET 処理をした値

PowerDownSelect プロパティ

概要

Power Down Select の値を取得または設定します。

処理内容

Read メソッド時に保持されます。

Write メソッドで PowerDownSelect 値を使用して書込みます。

`CommandType.FastMode`: DAC レジスタを変更する。EEPROM は影響を受けません

`CommandType.DACRegister`: コンフィギュレーションビットとデータコードを DAC レジスタにロードする

`CommandType.DACRegisterAndEEPROM`: コンフィギュレーションビットとデータコードを DAC レジスタにロードし、EEPROM に書き込む

SET 処理

value 値をセットする。

GET 処理

Read メソッド時に保持した値

または、SET 処理をした値

register_Data プロパティ

概要

DAC Input Data (12 bits)の値を取得または設定します。

処理内容

Read メソッド時に保持されます。

Write メソッドで register_Data 値を使用して書込みます。

`CommandType.FastMode`: DAC レジスタを変更する。EEPROM は影響を受けません

`CommandType.DACRegister`: コンフィギュレーションビットとデータコードを DAC レジスタにロードする

`CommandType.DACRegisterAndEEPROM`: コンフィギュレーションビットとデータコードを DAC レジスタにロードし、EEPROM に書き込む

SET 処理

value 値をセットする。

GET 処理

Read メソッド時に保持した値

または、SET 処理をした値

EEPROM_PowerDownSelect プロパティ

概要

EEPROM MEMORY の Power-DownSelect の値を取得または設定します。

処理内容

Read メソッドを指定した時に保持されます。

SET 処理

なし

GET 処理

Read メソッド時に保持した値

EEPROM_Data プロパティ

概要

EEPROM MEMORY の Data (12 bits)の値を取得または設定します。

処理内容

Read メソッドを指定した時に保持されます。

SET 処理

なし

GET 処理

Read メソッド時に保持した値

Init メソッド

概要

I2C バスコントローラに指定された接続設定を使って非同期的に接続します。

引数

| 引数名 | 型 | 説明 |
|---------|------|-----------|
| Address | byte | I2C のアドレス |

戻り値

なし

Read メソッド

概要

最新の変換結果を読み込みます。

引数

| 引数名 | 型 | 説明 |
|-----|---|----|
| | | |

戻り値

Status と同じ

1: 正常

1 以外: 異常

読込処理

| 引数 | 読込プロパティ |
|----|--|
| なし | RDYBSY POR PowerDownSelect register_Data EEPROM_PowerDownSelect EEPROM_Data |

Write メソッド

概要

引数 RegSelect に関するプロパティを編集し書き込みます。

引数

| 引数名 | 型 | 説明 |
|-------------|------|---|
| CommandType | byte | FastMode , DACRegister , DACRegisterAndEEPROM |

戻り値

Status

1: 正常

1 以外: 異常

書込処理

| CommandType 引数 | 書込プロパティ |
|----------------|----------------------------------|
| FastMode | PowerDownSelect register_Data |

サンプルプログラム

```
Imports I2CLibrary
Public NotInheritable Class MCP4725Page
    Inherits Page

    Private Const MCP4725_Address As Byte = &H60
    Private MCP4725 As MCP4725 = New MCP4725()

    Public Sub New()
        InitializeComponent()
        Call Init()
    End Sub

    Private Async Sub Init()
        Await MCP4725.Init(MCP4725_Address)
    End Sub

    Private Sub btnFirst_Click(sender As Object, e As RoutedEventArgs)
        MCP4725.PowerDownSelect = MCP4725.power_down_mode.Normal
        MCP4725.register_Data = 1000
        MCP4725.Write(MCP4725.CommandType.FastMode)
    End Sub

    Private Sub btnDACREG_Click(sender As Object, e As RoutedEventArgs)
        MCP4725.PowerDownSelect = MCP4725.power_down_mode.Normal
        MCP4725.register_Data = 2000
        MCP4725.Write(MCP4725.CommandType.DACRegister)
    End Sub

    Private Sub btnEEPROM_Click(sender As Object, e As RoutedEventArgs)
        MCP4725.PowerDownSelect = MCP4725.power_down_mode.Normal
        MCP4725.register_Data = 3000
        MCP4725.Write(MCP4725.CommandType.DACRegisterAndEEPROM)
    End Sub

    Private Sub btnRead_Click(sender As Object, e As RoutedEventArgs)
        MCP4725.Read()
        Debug.WriteLine("RDYBSY = " & MCP4725.RDYBSY.ToString)
        Debug.WriteLine("POR = " & MCP4725.POR.ToString)
        Debug.WriteLine("PowerDownSelect = " & MCP4725.PowerDownSelect.ToString)
        Debug.WriteLine("register_Data = " & MCP4725.register_Data.ToString)
        Debug.WriteLine("EEPROM_PowerDownSelect = " & MCP4725.EEPROM_PowerDownSelect.ToString)
        Debug.WriteLine("EEPROM_Data = " & MCP4725.EEPROM_Data.ToString)
    End Sub

    Private Sub btnBack_Click(sender As Object, e As RoutedEventArgs)
        MCP4725.close()
        App.rootFrame.Navigate(GetType(MenuPage))
    End Sub

End Class
```

3-6.気温・湿度・気圧センサ BME280

変更履歴

2016年9月15日 第一版 新規作成

1. 概要

気圧・温度・湿度センサ(BME280)と I2C 接続するための機能を提供します。

2. 機能一覧

プロパティ

| 機能名 | 機能説明 |
|-----------|----------------------------------|
| hum | 湿度の測定出力データを取得します。 |
| temp | 温度の測定出力データを取得します。 |
| press | 気圧の測定出力データを取得します。 |
| t_sb | ノーマルモードでの非アクティブな継続時間を取得または設定します。 |
| filter | IIR フィルタの時定数を取得または設定します。 |
| spi3w_en | 3 線式 SPI インターフェイスの設定を取得または設定します。 |
| osrs_t | 温度データのオーバーサンプリング制御を取得または設定します。 |
| osrs_p | 気圧データのオーバーサンプリング制御を取得または設定します。 |
| mode | デバイスのセンサーモードを取得または設定します。 |
| measuring | 測定状態を取得します。 |
| im_update | レジスタ更新状態を取得します。 |
| osrs_h | 湿度データのオーバーサンプリング制御を取得または設定します。 |

| 機能名 | 機能説明 |
|--------|--------------------------|
| Reset | パワーオンリセット手順を使用してリセットします。 |
| ID | チップ識別番号を取得します。 |
| dig_T1 | 温度補正関連値 1 を取得します。 |
| dig_T2 | 温度補正関連値 2 を取得します。 |
| dig_T3 | 温度補正関連値 3 を取得します。 |
| dig_P1 | 気圧補正関連値 1 を取得します。 |
| dig_P2 | 気圧補正関連値 2 を取得します。 |
| dig_P3 | 気圧補正関連値 3 を取得します。 |
| dig_P4 | 気圧補正関連値 4 を取得します。 |
| dig_P5 | 気圧補正関連値 5 を取得します。 |
| dig_P6 | 気圧補正関連値 6 を取得します。 |
| dig_P7 | 気圧補正関連値 7 を取得します。 |
| dig_P8 | 気圧補正関連値 8 を取得します。 |
| dig_P9 | 気圧補正関連値 9 を取得します。 |
| dig_H1 | 湿度補正関連値 1 を取得します。 |
| dig_H2 | 湿度補正関連値 2 を取得します。 |
| dig_H3 | 湿度補正関連値 3 を取得します。 |
| dig_H4 | 湿度補正関連値 4 を取得します。 |
| dig_H5 | 湿度補正関連値 5 を取得します。 |
| dig_H6 | 湿度補正関連値 6 を取得します。 |

メソッド

| 機能名 | 機能説明 |
|-----------------|-----------------------|
| Init | 初期化処理を行います。 |
| Read | 指定レジスタ読込 |
| ReadTemperature | 各プロパティの値を使用し温度を計算します。 |
| ReadPreasure | 各プロパティの値を使用し気圧を計算します。 |
| ReadHumidity | 各プロパティの値を使用し湿度を計算します。 |
| Write | 指定レジスタ書込 |

定数

| 機能名 | 機能説明 | 定数値 |
|--------------------------|--------------------|------|
| REG_ADDR.HUM | 湿度の測定出力データ | 0xFD |
| REG_ADDR.TEMP | 温度の測定出力データ | 0xFA |
| REG_ADDR.PRESS | 気圧の測定出力データ | 0xF7 |
| REG_ADDR.CONFIG | コンフィグ | 0xF5 |
| REG_ADDR.CTRL_MEAS | 計測制御 | 0xF4 |
| REG_ADDR.STATUS | 状態 | 0xF3 |
| REG_ADDR.CTRL_HUM | 湿度の測定出力データ | 0xF2 |
| REG_ADDR.CALIB26_CALIB41 | キャリブレーションデータ 26～41 | 0xE1 |
| REG_ADDR.RESET | リセット | 0xE0 |
| REG_ADDR.ID | ID | 0xD0 |
| REG_ADDR.CALIB00_CALIB25 | キャリブレーションデータ 00～25 | 0x88 |

hum プロパティ

概要

湿度の測定出力データを取得します。

処理内容

Read メソッドで `REG_ADDR.HUM` を指定した時に保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.HUM` を指定した時に保持した値
または、SET 処理をした値

temp プロパティ

概要

温度の測定出力データを取得します。

処理内容

Read メソッドで `REG_ADDR.TEMP` を指定した時に保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.TEMP` を指定した時に保持した値
または、SET 処理をした値

press プロパティ

概要

気圧の測定出力データを取得します。

処理内容

Read メソッドで `REG_ADDR.PRESS` を指定した時に保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.PRESS` を指定した時に保持した値
または、SET 処理をした値

t_sb プロパティ

概要

ノーマルモードでの非アクティブな継続時間を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.CONFIG` を指定した時に保持されます。

Write メソッドで `REG_ADDR.CONFIG` を指定した時に `t_sb` 値を使用して書込みます。

0 : 0.5 ms
1 : 62.5 ms
2 : 125 ms
3 : 250 ms
4 : 500 ms
5 : 1000 ms
6 : 10 ms
7 : 20 ms

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.CONFIG` を指定した時に保持した値
または、SET 処理をした値

filter プロパティ

概要

IIR フィルタの時定数を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.CONFIG` を指定した時に保持されます。

Write メソッドで `REG_ADDR.CONFIG` を指定した時に filter 値を使用して書込みます。

フィルタ係数
0 : filter off
1 : 2
2 : 4
3 : 8
4 : 16

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.CONFIG` を指定した時に保持した値
または、SET 処理をした値

spi3w_en プロパティ

概要

3 線式 SPI インターフェイスの設定を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.CONFIG` を指定した時に保持されます。

Write メソッドで `REG_ADDR.CONFIG` を指定した時に `spi3w_en` 値を使用して書込みます。

1 : 3 線式 SPI インターフェイス有効

SET 処理

`value` 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.CONFIG` を指定した時に保持した値

または、SET 処理をした値

osrs_t プロパティ

概要

温度データのオーバサンプリング制御を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.CTRL_MEAS` を指定した時に保持されます。

Write メソッドで `REG_ADDR.CTRL_MEAS` を指定した時に `osrs_t` 値を使用して書込みます。

- 0 : Skipped
- 1 : oversampling ×1
- 2 : oversampling ×2
- 3 : oversampling ×4
- 4 : oversampling ×8
- 5 : oversampling ×16

SET 処理

`value` 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.CTRL_MEAS` を指定した時に保持した値
または、SET 処理をした値

osrs_p プロパティ

概要

気圧データのオーバサンプリング制御を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.CTRL_MEAS` を指定した時に保持されます。

Write メソッドで `REG_ADDR.CTRL_MEAS` を指定した時に `osrs_p` 値を使用して書込みます。

- 0 : Skipped
- 1 : oversampling ×1
- 2 : oversampling ×2
- 3 : oversampling ×4
- 4 : oversampling ×8
- 5 : oversampling ×16

SET 処理

`value` 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.CTRL_MEAS` を指定した時に保持した値
または、SET 処理をした値

mode プロパティ

概要

デバイスのセンサーモードを取得または設定します。

処理内容

Read メソッドで `REG_ADDR.CTRL_MEAS` を指定した時に保持されます。

Write メソッドで `REG_ADDR.CTRL_MEAS` を指定した時に `mode` 値を使用して書込みます。

0 : SleepMode

1,2 : Forced mode

3 : Normal mode

SET 処理

`value` 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.CTRL_MEAS` を指定した時に保持した値

または、SET 処理をした値

measuring プロパティ

概要

測定状態を取得します。

処理内容

Read メソッドで `REG_ADDR.STATUS` を指定した時に保持されます。

コンバージョンが実行されているときは常に自動的に '1' に設定されます
結果が転送されると「0」に戻ります

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.STATUS` を指定した時に保持した値
または、SET 処理をした値

im_update プロパティ

概要

測定状態を取得します。

処理内容

Read メソッドで `REG_ADDR.STATUS` を指定した時に保持されます。

NVM データが存在するときに自動的に '1' に設定され、
イメージレジスタにコピーされ、'0' に戻ります。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.STATUS` を指定した時に保持した値
または、SET 処理をした値

osrs_h プロパティ

概要

湿度データのオーバサンプリング制御を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.CTRL_HUM` を指定した時に保持されます。

Write メソッドで `REG_ADDR.CTRL_HUM` を指定した時に `osrs_h` 値を使用して書込みます。

- 0 : Skipped
- 1 : oversampling ×1
- 2 : oversampling ×2
- 3 : oversampling ×4
- 4 : oversampling ×8
- 5 : oversampling ×16

SET 処理

`value` 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.CTRL_HUM` を指定した時に保持した値
または、SET 処理をした値

Reset プロパティ

概要

パワーオンリセット手順を使用してリセットします。

処理内容

Write メソッドで `REG_ADDR.RESET` を指定した時に Reset 値を使用して書込みます。

書き込み値: 0xB6

SET 処理

value 値をセットする。

GET 処理

SET 処理をした値

ID プロパティ

概要

チップ識別番号を取得します。

処理内容

Read メソッドで `REG_ADDR.ID` を指定した時に保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.ID` を指定した時に保持した値
または、SET 処理をした値

dig_T1 プロパティ

概要

温度補正関連値 1 を取得します。

処理内容

Read メソッドで `REG_ADDR.CALIB00_CALIB25` を指定した時に保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.CALIB00_CALIB25` を指定した時に保持した値
または、SET 処理をした値

dig_T2 プロパティ

概要

温度補正関連値 2 を取得します。

処理内容

Read メソッドで `REG_ADDR.CALIB00_CALIB25` を指定した時に保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.CALIB00_CALIB25` を指定した時に保持した値
または、SET 処理をした値

dig_T3 プロパティ

概要

温度補正関連値 3 を取得します。

処理内容

Read メソッドで `REG_ADDR.CALIB00_CALIB25` を指定した時に保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.CALIB00_CALIB25` を指定した時に保持した値
または、SET 処理をした値

dig_P1 プロパティ

概要

気圧補正関連値 1 を取得します。

処理内容

Read メソッドで `REG_ADDR.CALIB00_CALIB25` を指定した時に保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.CALIB00_CALIB25` を指定した時に保持した値
または、SET 処理をした値

dig_P2 プロパティ

概要

気圧補正関連値 2 を取得します。

処理内容

Read メソッドで `REG_ADDR.CALIB00_CALIB25` を指定した時に保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.CALIB00_CALIB25` を指定した時に保持した値
または、SET 処理をした値

dig_P3 プロパティ

概要

気圧補正関連値 3 を取得します。

処理内容

Read メソッドで `REG_ADDR.CALIB00_CALIB25` を指定した時に保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.CALIB00_CALIB25` を指定した時に保持した値
または、SET 処理をした値

dig_P4 プロパティ

概要

気圧補正関連値 4 を取得します。

処理内容

Read メソッドで `REG_ADDR.CALIB00_CALIB25` を指定した時に保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.CALIB00_CALIB25` を指定した時に保持した値
または、SET 処理をした値

dig_P5 プロパティ

概要

気圧補正関連値 5 を取得します。

処理内容

Read メソッドで `REG_ADDR.CALIB00_CALIB25` を指定した時に保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.CALIB00_CALIB25` を指定した時に保持した値
または、SET 処理をした値

dig_P6 プロパティ

概要

気圧補正関連値 6 を取得します。

処理内容

Read メソッドで `REG_ADDR.CALIB00_CALIB25` を指定した時に保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.CALIB00_CALIB25` を指定した時に保持した値
または、SET 処理をした値

dig_P7 プロパティ

概要

気圧補正関連値 7 を取得します。

処理内容

Read メソッドで `REG_ADDR.CALIB00_CALIB25` を指定した時に保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.CALIB00_CALIB25` を指定した時に保持した値
または、SET 処理をした値

dig_P8 プロパティ

概要

気圧補正関連値 8 を取得します。

処理内容

Read メソッドで `REG_ADDR.CALIB00_CALIB25` を指定した時に保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.CALIB00_CALIB25` を指定した時に保持した値
または、SET 処理をした値

dig_P9 プロパティ

概要

気圧補正関連値 9 を取得します。

処理内容

Read メソッドで `REG_ADDR.CALIB00_CALIB25` を指定した時に保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.CALIB00_CALIB25` を指定した時に保持した値
または、SET 処理をした値

dig_H1 プロパティ

概要

湿度補正関連値 1 を取得します。

処理内容

Read メソッドで `REG_ADDR.CALIB00_CALIB25` を指定した時に保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.CALIB00_CALIB25` を指定した時に保持した値
または、SET 処理をした値

dig_H2 プロパティ

概要

湿度補正関連値 2 を取得します。

処理内容

Read メソッドで `REG_ADDR.CALIB26_CALIB41` を指定した時に保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.CALIB26_CALIB41` を指定した時に保持した値
または、SET 処理をした値

dig_H3 プロパティ

概要

湿度補正関連値 3 を取得します。

処理内容

Read メソッドで `REG_ADDR.CALIB26_CALIB41` を指定した時に保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.CALIB26_CALIB41` を指定した時に保持した値
または、SET 処理をした値

dig_H4 プロパティ

概要

湿度補正関連値 4 を取得します。

処理内容

Read メソッドで `REG_ADDR.CALIB26_CALIB41` を指定した時に保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.CALIB26_CALIB41` を指定した時に保持した値
または、SET 処理をした値

dig_H5 プロパティ

概要

湿度補正関連値 5 を取得します。

処理内容

Read メソッドで `REG_ADDR.CALIB26_CALIB41` を指定した時に保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.CALIB26_CALIB41` を指定した時に保持した値
または、SET 処理をした値

dig_H6 プロパティ

概要

湿度補正関連値 6 を取得します。

処理内容

Read メソッドで `REG_ADDR.CALIB26_CALIB41` を指定した時に保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.CALIB26_CALIB41` を指定した時に保持した値
または、SET 処理をした値

Init メソッド

概要

初期化処理を行います。

処理内容

1. I2C デバイスの初期化処理を実行します。
2. 補正值データ読み込み処理を実行します。

引数

| 引数名 | 型 | 説明 |
|---------|------|-----------|
| address | byte | I2C のアドレス |

戻り値

なし

エラー

1:正常

Exception の場合 : lastException

Read メソッド

概要

最新の変換結果を読みます。

引数

| 引数名 | 型 | 説明 |
|----------|------|---|
| REG_ADDR | byte | REG_ADDR_CALIB00_CALIB25 (0x88) ~ REG_ADDR_HUM (0xED) |

戻り値

Status と同じ

1: 正常

1 以外: 異常

読込処理

| REG_ADDR 値 | 読込プロパティ |
|------------|----------------------------|
| HUM | hum |
| TEMP | temp |
| PRESS | press |
| CONFIG | spi3w_en filter t_sb |
| CTRL_MEAS | mode osrs_p osrs_t |
| STATUS | im_update measuring |
| CTRL_HUM | osrs_h |
| ID | ID |

| REG_ADDR 値 | 読込プロパティ |
|-----------------|--|
| CALIB00_CALIB25 | dig_T1 dig_T2 dig_T3 dig_P1 dig_P2 dig_P3 dig_P4 dig_P5 dig_P6 dig_P7 dig_P8 dig_P9 dig_H1 |
| CALIB26_CALIB41 | dig_H2 dig_H3 dig_H4 dig_H5 dig_H6 |

Write メソッド

概要

引数 RegSelect に関するプロパティを編集し書き込みます。

処理内容

1. Status = 0
2. RegSelect によって書き込むデータを編集
3. I2cDeviceID.Write
4. Status = 1
5. Exception が発生した場合は、
throw 再スローします。

引数

| 引数名 | 型 | 説明 |
|----------|------|----|
| REG_ADDR | byte | |

戻り値

Status

- 1: 正常
- 1 以外: 異常

書込処理

| REG_ADDR 値 | 書込プロパティ |
|------------|----------------------------|
| RESET | Reset |
| CTRL_HUM | osrs_h |
| CTRL_MEAS | mode osrs_p osrs_t |
| CONFIG | spi3w_en filter t_sb |

ReadTemperature メソッド

概要

気温データの Raw 値から補正値を使用して気温を計算します。(単位: degC)

処理内容

```
var1 = (temp / 16384.0 - dig_T1 / 1024.0) * dig_T2;  
var2 = ((temp / 131072.0 - dig_T1 / 8192.0) *  
        (temp / 131072.0 - dig_T1 / 8192.0)) * dig_T3;
```

```
t_fine = (int)(var1 + var2);  
気温 = (var1 + var2) / 5120.0;
```

引数

なし

戻り値

補正後の値(気温)

エラー

1:正常

Exception の場合: lastException

ReadPressure メソッド

概要

気圧データの Raw 値から補正値を使用して気圧を計算します。(単位:%)

処理内容

```

var1 = (long)t_fine - 128000;
var2 = var1 * var1 * dig_P6;
var2 = var2 + ((var1 * dig_P5) << 17);
var2 = var2 + ((long)dig_P4 << 35);
var1 = ((var1 * var1 * dig_P3) >> 8) + ((var1 * dig_P2) << 12);
var1 = (((long)1 << 47) + var1) * dig_P1 >> 33;
if (var1 == 0)
{
    return 0; //Avoid exception caused by division by zero
}
//Perform calibration operations as per datasheet:
p = 1048576 - press;
p = (((p << 31) - var2) * 3125) / var1;
var1 = ((long)dig_P9 * (p >> 13) * (p >> 13)) >> 25;
var2 = ((long)dig_P8 * p) >> 19;
p = ((p + var1 + var2) >> 8) + ((long)dig_P7 << 4);

    気圧 = ((float) p) / 256
  
```

引数

なし

戻り値

補正後の値(気圧)

エラー

1:正常

Exception の場合 : lastException

ReadHumidity メソッド

概要

湿度データの Raw 値から補正値を使用して気温を計算します。(単位: Pa)

処理内容

```
var_H = t_fine - 76800.0;
var_H = (hum - (dig_H4 * 64.0 + dig_H5 / 16384.0 * var_H)) *
        dig_H2 / 65536.0 * (1.0 + dig_H6 / 67108864.0 * var_H *
        (1.0 + dig_H3 / 67108864.0 * var_H));
var_H = var_H * (1.0 - dig_H1 * var_H / 524288.0);

if (var_H > 100.0)
{
    var_H = 100.0;
}

else if (var_H < 0.0)
{
    var_H = 0.0;
}

湿度 = var_H;
```

引数

なし

戻り値

補正後の値(湿度)

エラー

1:正常

Exception の場合 : lastException

サンプルプログラム

```
Imports I2CLibrary
Public NotInheritable Class BME280Page
    Inherits Page

    Private Const BME280_Address As Byte = &H77
    Private BME280 As BME280 = New BME280()
    Private WithEvents RefreshTimer As DispatcherTimer = New DispatcherTimer()

    Public Sub New()
        InitializeComponent()
        Call Init()
    End Sub

    Private Async Sub Init()
        Await BME280.Init(BME280_Address)
    End Sub

    Private Sub btnStart_Click(sender As Object, e As RoutedEventArgs)
        BME280.Read(BME280.REG_ADDR.CONFIG)
        BME280.Read(BME280.REG_ADDR.CTRL_MEAS)
        BME280.Read(BME280.REG_ADDR.CTRL_HUM)

        BME280.t_sb = BME280.standbySettings_e.tsb_0p5ms
        BME280.mode = BME280.mode_e.smNormal
        BME280.filter = BME280.filterCoefficient_e.fc_16
        BME280.osrs_p = BME280.oversampling_e.os16x
        BME280.osrs_t = BME280.oversampling_e.os16x
        BME280.osrs_h = BME280.oversampling_e.os16x

        BME280.Write(BME280.REG_ADDR.CTRL_HUM)
        BME280.Write(BME280.REG_ADDR.CTRL_MEAS)
        BME280.Write(BME280.REG_ADDR.CONFIG)

        RefreshTimer.Interval = TimeSpan.FromMilliseconds(100)
        RefreshTimer.Start()
    End Sub

    Private Sub Measurement(sender As Object, e As Object) Handles RefreshTimer.Tick
        RefreshTimer.Stop()

        BME280.Read(BME280.REG_ADDR.HUM)
        BME280.Read(BME280.REG_ADDR.TEMP)
        BME280.Read(BME280.REG_ADDR.PRESS)
        BME280.Read(BME280.REG_ADDR.CONFIG)
        BME280.Read(BME280.REG_ADDR.CTRL_MEAS)
        BME280.Read(BME280.REG_ADDR.STATUS)
        BME280.Read(BME280.REG_ADDR.CTRL_HUM)
        BME280.Read(BME280.REG_ADDR.CALIB26_CALIB41)
        BME280.Read(BME280.REG_ADDR.RESET)
        BME280.Read(BME280.REG_ADDR.ID)
        BME280.Read(BME280.REG_ADDR.CALIB00_CALIB25)
    End Sub
End Class
```

```
Dim ftmp As Single = BME280.ReadTemperature
Dim fhum As Single = BME280.ReadHumidity()
Dim fpres As Single = BME280.ReadPreasure() / 100

Debug.WriteLine("[0:F2] °C", ftmp)
Debug.WriteLine("[0:F2] %", fhum)
Debug.WriteLine("[0:F2] hPa", fpres)

RefreshTimer.Start()
End Sub

Private Sub btnStop_Click(sender As Object, e As RoutedEventArgs)
    RefreshTimer.Stop()
End Sub

Private Sub btnBack_Click(sender As Object, e As RoutedEventArgs)
    RefreshTimer.Stop()
    BME280.close()
    App.rootFrame.Navigate(GetType(MenuPage))
End Sub

End Class
```

3-7.ステッピングモータコントローラ PCA9629A

変更履歴

2016年11月25日 第一版 新規作成

1. 概要

ステッピングモータードライバIF設計書 (PCA9629A) と I2C 接続するための機能を提供します。

2. 機能一覧

プロパティ

| 機能名 | 機能説明 |
|------------|--|
| MODE | Mode register |
| WDTOI | WDCNTL [2:1] Watchdog time-out interval register |
| WDMOD | WDCNTL[0:0] ウォッチドッグ割り込みとモータ停止モード。 |
| WDEN | IO_CFG[7:4] ウォッチドッグを有効にします。 |
| P0_CFG | IO_CFG[0:0] P0 入出力設定 |
| P1_CFG | IO_CFG[1:1] P1 入出力設定 |
| P2_CFG | IO_CFG[2:2] P2 入出力設定 |
| P3_CFG | IO_CFG[3:3] P3 入出力設定 |
| P0_INTMODE | INTMODE[0:0] P0 割込設定 |
| P1_INTMODE | INTMODE[1:1] P1 割込設定 |
| P2_INTMODE | INTMODE[2:2] P2 割込設定 |
| P3_INTMODE | INTMODE[3:3] P3 割込設定 |

| 機能名 | 機能説明 |
|---------------------|---|
| P0P1_FILTER | INTMODE[6:4] pulse width of spike or noise that must be suppressed by the input filter for P0 and P1 inputs |
| P0_MSK | MSK [0:0] P0 に対する割込の有効／無効 |
| P1_MSK | MSK [1:1] P1 に対する割込の有効／無効 |
| P2_MSK | MSK [2:2] P2 に対する割込の有効／無効 |
| P3_MSK | MSK [3:3] P3 に対する割込の有効／無効 |
| MOTOR_MSK | MSK [4:4] モータ停止時の割り込み有効／無効 |
| INTP0_FLAG | INTSTAT [0:0] INTP0 の割込ステータス |
| INTP1_FLAG | INTSTAT [1:1] INTP1 の割込ステータス |
| INTP2_FLAG | INTSTAT [2:2] INTP2 の割込ステータス |
| INTP3_FLAG | INTSTAT [3:3] INTP3 の割込ステータス |
| MOTOR_STOP_FLAG | INTSTAT [4:4] Motor stop の割込ステータス |
| WDINT_WATCHDOG_FLAG | INTSTAT [5:5] WDINT watchdog の割込ステータス |
| IP | I / O P0～P3 の入力論理レベルを反映します |
| MOTOR_CONTROL | INT_MTR_ACT[7:5] motor on INT |
| INT_AUTO | INT_MTR_ACT[4:3] |
| INT_MOTOR | INT_MTR_ACT[0] enable interrupt based control of motor |
| EXTRASTEPS0 | EXTRASTEPS のカウント値(ステップ)INTP0 の場合 |
| EXTRASTEPS1 | EXTRASTEPS のカウント値(ステップ)INTP1 の場合 |

| 機能名 | 機能説明 |
|------------------------------|---|
| PH_CNTL | OP_CFG_PHS[7:6] 単相ドライブ 2相ドライブ ハーフステップドライブ出力 |
| OUT_CNTL | OP_CFG_PHS[4:4] モータ駆動、汎用出力 |
| OP_LEVELS | OP_CFG_PHS[3:0] 出力論理レベル |
| Motor_stop_timer | OP_STAT_TO[7:5] モータ停止タイムアウトタイマ |
| CCW_stop | OP_STAT_TO[3:2] CCW stop |
| CW_stop | OP_STAT_TO[1:0] CW stop |
| RUCNTL_enable | RUCNTL[5:5] 開始時にランプアップ |
| RUCNTL_change | RUCNTL[4:4] ランプアップを再度有効にしてランプアップ率を変更する |
| RUCNTL_Multiplication_factor | RUCNTL[3:0] ランプアップステップ増倍率 |
| RDCNTL_enable | RDCNTL [5:5] 停止時にランプダウン |
| RDCNTL_change | RDCNTL [4:4] ランプダウンを再び有効にしてランプダウン率を変更する |
| RDCNTL_Multiplication_factor | RDCNTL [3:0] ランプダウンステップ増倍率 |
| PMA | 複数のアクションを制御する |
| LOOPDLY_CW | 反転するためのループ遅延カウンタ 時計回り→反時計回り |
| LOOPDLY_CCW | 反転するためのループ遅延カウンタ 反時計回りから時計回り |
| CWSCOUNT | 時計回りのステップ数 |
| CCWSCOUNT | 反時計回りのステップ数 |
| CWPWL_PRESCALER | 時計回りのステップパルス幅 PRESCALER |

| 機能名 | 機能説明 |
|-------------------------|--------------------------|
| CWPWL_STEP_PULSE_WIDTH | 時計回りのステップパルス幅 |
| CCWPWL_PRESCALER | 反時計回りのステップパルス幅 PRESCALER |
| CCWPWL_STEP_PULSE_WIDTH | 反時計回りのステップパルス幅 |
| MCNTL | Motor control |
| SUBADR1 | I2C-bus subaddress1 |
| SUBADR2 | I2C-bus subaddress2 |
| SUBADR3 | I2C-bus subaddress3 |
| ALLCALLADR | All Call I2C-bus address |
| STEPCOUNT | Step counter |
| ReadBuff | 読込生データ |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |

メソッド

| 機能名 | 機能説明 |
|-------|---------------------------------------|
| Init | I2C バスコントローラに指定された接続設定を使って非同期的に接続します。 |
| Read | 指定レジスタ読込 |
| Write | 指定レジスタ書込 |

定数

| 機能名 | 機能説明 | 定数值 |
|----------------------|---|------|
| REG_ADDR.MODE | Mode register | 0x00 |
| REG_ADDR.WDIOI | Watchdog time-out interval register | 0x01 |
| REG_ADDR.WDCNTL | Watchdog control register | 0x02 |
| REG_ADDR.IO_CFG | I/O Configuration register | 0x03 |
| REG_ADDR.INTMODE | Interrupt mode register | 0x04 |
| REG_ADDR.MSK | Mask interrupt register | 0x05 |
| REG_ADDR.INTSTAT | Interrupt status register | 0x06 |
| REG_ADDR.IP | Input port register | 0x07 |
| REG_ADDR.INT_MTR_ACT | Interrupt motor action control register | 0x08 |
| REG_ADDR.EXTRASTEPS0 | Count value for extra steps for INTP0 | 0x09 |
| REG_ADDR.EXTRASTEPS1 | Count value for extra steps for INTP1 | 0x0A |
| REG_ADDR.OP_CFG_PHS | Output port configuration and phase control register | 0x0B |
| REG_ADDR.OP_STAT_TO | Output port state and time-out control register | 0x0C |
| REG_ADDR.RUCNTL | Ramp up control register | 0x0D |
| REG_ADDR.RDCNTL | Ramp down control register | 0x0E |
| REG_ADDR.PMA | Perform multiple of actions control register | 0x0F |
| REG_ADDR.LOOPDLY_CW | Loop delay time for reversing from CW to CCW register | 0x10 |
| REG_ADDR.LOOPDLY_CCW | Loop delay time for reversing from CCW to CW register | 0x11 |

| 機能名 | 機能説明 | 定数值 |
|---------------------|---|------|
| REG_ADDR.CWSCOUNTL | Number of steps CW low byte | 0x12 |
| REG_ADDR.CWSCOUNTH | Number of steps CW high byte | 0x13 |
| REG_ADDR.CCWSCOUNTL | Number of steps CCW low byte | 0x14 |
| REG_ADDR.CCWSCOUNTH | Number of steps CCW high byte | 0x15 |
| REG_ADDR.CWPWL | Step pulse width for CW rotation low byte | 0x16 |
| REG_ADDR.CWPWH | Step pulse width for CW rotation high byte | 0x17 |
| REG_ADDR.CCWPWL | Step pulse width for CCW rotation low byte | 0x18 |
| REG_ADDR.CCWPWH | Step pulse width for CCW rotation high byte | 0x19 |
| REG_ADDR.MCNTL | Motor start/stop and rotate direction control | 0x1A |
| REG_ADDR.SUBADR1 | I2C-bus subaddress 1 | 0x1B |
| REG_ADDR.SUBADR2 | I2C-bus subaddress 2 | 0x1C |
| REG_ADDR.SUBADR3 | I2C-bus subaddress 3 | 0x1D |
| REG_ADDR.ALLCALLADR | All Call I2C-bus address | 0x1E |
| REG_ADDR.STEPCOUNT0 | Step counter byte 0 | 0x1F |
| REG_ADDR.STEPCOUNT1 | Step counter byte 1 | 0x20 |
| REG_ADDR.STEPCOUNT2 | Step counter byte 2 | 0x21 |
| REG_ADDR.STEPCOUNT3 | Step counter byte 3 | 0x22 |
| | | |
| Auto_Increment | Auto-Increment | 0x80 |

MODE プロパティ

概要

MODE レジスタの値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.MODE` を指定した時に保持されます。

Write メソッドで `REG_ADDR.MODE` を指定した時に MODE 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.MODE` を指定した時に保持した値
または、SET 処理をした値

WDTOI プロパティ

概要

WDTOI レジスタの WDTOI の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.WDTOI` を指定した時に保持されます。

Write メソッドで `REG_ADDR.WDTOI` を指定した時に WDTOI 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.WDTOI` を指定した時に保持した値
または、SET 処理をした値

WDMOD プロパティ

概要

WDCNTL レジスタの WDMOD の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.WDCNTL` を指定した時に保持されます。

Write メソッドで `REG_ADDR.WDCNTL` を指定した時に WDMOD 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.WDCNTL` を指定した時に保持した値
または、SET 処理をした値

WDMOD プロパティ

概要

WDCNTL レジスタの WDMOD の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.WDCNTL` を指定した時に保持されます。

Write メソッドで `REG_ADDR.WDCNTL` を指定した時に WDMOD 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.WDCNTL` を指定した時に保持した値
または、SET 処理をした値

WDEN プロパティ

概要

IO_CFG レジスタの WDEN の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.IO_CFG` を指定した時に保持されます。

Write メソッドで `REG_ADDR.IO_CFG` を指定した時に WDEN 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.IO_CFG` を指定した時に保持した値
または、SET 処理をした値

P0_CFG プロパティ

概要

IO_CFG レジスタの P0_CFG の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.IO_CFG` を指定した時に保持されます。

Write メソッドで `REG_ADDR.IO_CFG` を指定した時に P0_CFG 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.IO_CFG` を指定した時に保持した値
または、SET 処理をした値

P1_CFG プロパティ

概要

IO_CFG レジスタの P1_CFG の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.IO_CFG` を指定した時に保持されます。

Write メソッドで `REG_ADDR.IO_CFG` を指定した時に P1_CFG 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.IO_CFG` を指定した時に保持した値
または、SET 処理をした値

P2_CFG プロパティ

概要

IO_CFG レジスタの P2_CFG の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.IO_CFG` を指定した時に保持されます。

Write メソッドで `REG_ADDR.IO_CFG` を指定した時に P2_CFG 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.IO_CFG` を指定した時に保持した値
または、SET 処理をした値

P3_CFG プロパティ

概要

IO_CFG レジスタの P3_CFG の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.IO_CFG` を指定した時に保持されます。

Write メソッドで `REG_ADDR.IO_CFG` を指定した時に P3_CFG 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.IO_CFG` を指定した時に保持した値
または、SET 処理をした値

P0_INTMODE プロパティ

概要

INTMODE レジスタの P0_INTMODE の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.INTMODE` を指定した時に保持されます。

Write メソッドで `REG_ADDR.INTMODE` を指定した時に P0_INTMODE 値を使用して書込ま
す。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.INTMODE` を指定した時に保持した値
または、SET 処理をした値

P1_INTMODE プロパティ

概要

INTMODE レジスタの P1_INTMODE の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.INTMODE` を指定した時に保持されます。

Write メソッドで `REG_ADDR.INTMODE` を指定した時に P1_INTMODE 値を使用して書込ま
す。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.INTMODE` を指定した時に保持した値
または、SET 処理をした値

P2_INTMODE プロパティ

概要

INTMODE レジスタの P2_INTMODE の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.INTMODE` を指定した時に保持されます。

Write メソッドで `REG_ADDR.INTMODE` を指定した時に P2_INTMODE 値を使用して書込み
す。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.INTMODE` を指定した時に保持した値
または、SET 処理をした値

P3_INTMODE プロパティ

概要

INTMODE レジスタの P3_INTMODE の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.INTMODE` を指定した時に保持されます。

Write メソッドで `REG_ADDR.INTMODE` を指定した時に P3_INTMODE 値を使用して書込み
す。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.INTMODE` を指定した時に保持した値
または、SET 処理をした値

POP1_FILTER プロパティ

概要

INTMODE レジスタの POP1_FILTER の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.INTMODE` を指定した時に保持されます。

Write メソッドで `REG_ADDR.INTMODE` を指定した時に POP1_FILTER 値を使用して書込み
す。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.INTMODE` を指定した時に保持した値
または、SET 処理をした値

P0_MSK プロパティ

概要

MSK レジスタの P0_MSK の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.MSK` を指定した時に保持されます。

Write メソッドで `REG_ADDR.MSK` を指定した時に P0_MSK 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.MSK` を指定した時に保持した値
または、SET 処理をした値

P1_MSK プロパティ

概要

MSK レジスタの P1_MSK の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.MSK` を指定した時に保持されます。

Write メソッドで `REG_ADDR.MSK` を指定した時に P1_MSK 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.MSK` を指定した時に保持した値
または、SET 処理をした値

P2_MSK プロパティ

概要

MSK レジスタの P2_MSK の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.MSK` を指定した時に保持されます。

Write メソッドで `REG_ADDR.MSK` を指定した時に P2_MSK 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.MSK` を指定した時に保持した値
または、SET 処理をした値

P3_MSK プロパティ

概要

MSK レジスタの P3_MSK の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.MSK` を指定した時に保持されます。

Write メソッドで `REG_ADDR.MSK` を指定した時に P3_MSK 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.MSK` を指定した時に保持した値
または、SET 処理をした値

MOTOR_MSK プロパティ

概要

MSK レジスタの MOTOR_MSK の値を取得または設定します。

処理内容

Read メソッドで REG_ADDR.MSK を指定した時に保持されます。

Write メソッドで REG_ADDR.MSK を指定した時に MOTOR_MSK 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで REG_ADDR.MSK を指定した時に保持した値
または、SET 処理をした値

INTP0_FLAG プロパティ

概要

INTSTAT レジスタの INTP0_FLAG の値を取得します。

処理内容

Read メソッドで `REG_ADDR.INTSTAT` を指定した時に保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.INTSTAT` を指定した時に保持した値
または、SET 処理をした値

INTP1_FLAG プロパティ

概要

INTSTAT レジスタの INTP1_FLAG の値を取得します。

処理内容

Read メソッドで `REG_ADDR.INTSTAT` を指定した時に保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.INTSTAT` を指定した時に保持した値
または、SET 処理をした値

INTP2_FLAG プロパティ

概要

INTSTAT レジスタの INTP2_FLAG の値を取得します。

処理内容

Read メソッドで `REG_ADDR.INTSTAT` を指定した時に保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.INTSTAT` を指定した時に保持した値
または、SET 処理をした値

INTP3_FLAG プロパティ

概要

INTSTAT レジスタの INTP3_FLAG の値を取得します。

処理内容

Read メソッドで `REG_ADDR.INTSTAT` を指定した時に保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.INTSTAT` を指定した時に保持した値
または、SET 処理をした値

MOTOR_STOP_FLAG プロパティ

概要

INTSTAT レジスタの INTP0_FLAG の値を取得します。

処理内容

Read メソッドで `REG_ADDR.INTSTAT` を指定した時に保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.INTSTAT` を指定した時に保持した値
または、SET 処理をした値

WDINT_WATCHDOG_FLAG プロパティ

概要

INTSTAT レジスタの WDINT_WATCHDOG_FLAG の値を取得します。

処理内容

Read メソッドで `REG_ADDR.INTSTAT` を指定した時に保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.INTSTAT` を指定した時に保持した値
または、SET 処理をした値

IP プロパティ

概要

IP レジスタの IP の値を取得します。

処理内容

Read メソッドで `REG_ADDR.IP` を指定した時に保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.IP` を指定した時に保持した値
または、SET 処理をした値

MOTOR_CONTROL プロパティ

概要

INT_MTR_ACT レジスタの MOTOR_CONTROL の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.INT_MTR_ACT` を指定した時に保持されます。

Write メソッドで `REG_ADDR.INT_MTR_ACT` を指定した時に MOTOR_CONTROL 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.INT_MTR_ACT` を指定した時に保持した値
または、SET 処理をした値

INT_AUTO プロパティ

概要

INT_MTR_ACT レジスタの INT_AUTO の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.INT_MTR_ACT` を指定した時に保持されます。

Write メソッドで `REG_ADDR.INT_MTR_ACT` を指定した時に INT_AUTO 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.INT_MTR_ACT` を指定した時に保持した値
または、SET 処理をした値

INT_MOTOR プロパティ

概要

INT_MTR_ACT レジスタの INT_MOTOR の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.INT_MTR_ACT` を指定した時に保持されます。

Write メソッドで `REG_ADDR.INT_MTR_ACT` を指定した時に INT_MOTOR 値を使用して書込ま
す。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.INT_MTR_ACT` を指定した時に保持した値
または、SET 処理をした値

EXTRASTEPS0 プロパティ

概要

EXTRASTEPS0 レジスタの EXTRASTEPS0 の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.EXTRASTEPS0` を指定した時に保持されます。

Write メソッドで `REG_ADDR.EXTRASTEPS0` を指定した時に EXTRASTEPS0 値を使用して書込まれます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.EXTRASTEPS0` を指定した時に保持した値
または、SET 処理をした値

EXTRASTEPS1 プロパティ

概要

EXTRASTEPS1 レジスタの EXTRASTEPS1 の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.EXTRASTEPS1` を指定した時に保持されます。

Write メソッドで `REG_ADDR.EXTRASTEPS1` を指定した時に EXTRASTEPS1 値を使用して書込まれます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.EXTRASTEPS1` を指定した時に保持した値
または、SET 処理をした値

OP_CFG_PHS プロパティ

概要

OP_CFG_PHS レジスタの PH_CNTL の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.OP_CFG_PHS` を指定した時に保持されます。

Write メソッドで `REG_ADDR.OP_CFG_PHS` を指定した時に PH_CNTL 値を使用して書込ます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.OP_CFG_PHS` を指定した時に保持した値
または、SET 処理をした値

OUT_CNTL プロパティ

概要

OP_CFG_PHS レジスタの OUT_CNTL の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.OP_CFG_PHS` を指定した時に保持されます。

Write メソッドで `REG_ADDR.OP_CFG_PHS` を指定した時に OUT_CNTL 値を使用して書込ます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.OP_CFG_PHS` を指定した時に保持した値
または、SET 処理をした値

OP_LEVELS プロパティ

概要

OP_CFG_PHS レジスタの OP_LEVELS の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.OP_CFG_PHS` を指定した時に保持されます。

Write メソッドで `REG_ADDR.OP_CFG_PHS` を指定した時に OP_LEVELS 値を使用して書込ます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.OP_CFG_PHS` を指定した時に保持した値
または、SET 処理をした値

Motor_stop_timer プロパティ

概要

OP_STAT_TO レジスタの Motor_stop_timer の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.OP_STAT_TO` を指定した時に保持されます。

Write メソッドで `REG_ADDR.OP_STAT_TO` を指定した時に Motor_stop_timer 値を使用して書込まれます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.OP_STAT_TO` を指定した時に保持した値
または、SET 処理をした値

CCW_stop プロパティ

概要

OP_STAT_TO レジスタの CCW_stop の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.OP_STAT_TO` を指定した時に保持されます。

Write メソッドで `REG_ADDR.OP_STAT_TO` を指定した時に CCW_stop 値を使用して書込
ます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.OP_STAT_TO` を指定した時に保持した値
または、SET 処理をした値

CW_stop プロパティ

概要

OP_STAT_TO レジスタの CW_stop の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.OP_STAT_TO` を指定した時に保持されます。

Write メソッドで `REG_ADDR.OP_STAT_TO` を指定した時に CW_stop 値を使用して書込ます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.OP_STAT_TO` を指定した時に保持した値
または、SET 処理をした値

RUCNTL_enable プロパティ

概要

RUCNTL レジスタの RUCNTL_enable の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.RUCNTL` を指定した時に保持されます。

Write メソッドで `REG_ADDR.RUCNTL` を指定した時に RUCNTL_enable 値を使用して書込まず。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.RUCNTL` を指定した時に保持した値
または、SET 処理をした値

RUCNTL_change プロパティ

概要

RUCNTL レジスタの RUCNTL_change の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.RUCNTL` を指定した時に保持されます。

Write メソッドで `REG_ADDR.RUCNTL` を指定した時に RUCNTL_change 値を使用して書込まず。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.RUCNTL` を指定した時に保持した値
または、SET 処理をした値

RUCNTL_Multiplication_factor プロパティ

概要

RUCNTL レジスタの RUCNTL_Multiplication_factor の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.RUCNTL` を指定した時に保持されます。

Write メソッドで `REG_ADDR.RUCNTL` を指定した時に RUCNTL_Multiplication_factor 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.RUCNTL` を指定した時に保持した値
または、SET 処理をした値

RDCNTL_enable プロパティ

概要

RDCNTL レジスタの RDCNTL_enable の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.RDCNTL` を指定した時に保持されます。

Write メソッドで `REG_ADDR.RDCNTL` を指定した時に RDCNTL_enable 値を使用して書込まず。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.RDCNTL` を指定した時に保持した値
または、SET 処理をした値

RDCNTL_change プロパティ

概要

RDCNTL レジスタの RDCNTL_change の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.RDCNTL` を指定した時に保持されます。

Write メソッドで `REG_ADDR.RDCNTL` を指定した時に RDCNTL_change 値を使用して書込まず。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.RDCNTL` を指定した時に保持した値
または、SET 処理をした値

RDCNTL_Multiplication_factor プロパティ

概要

RDCNTL レジスタの RDCNTL_Multiplication_factor の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.RDCNTL` を指定した時に保持されます。

Write メソッドで `REG_ADDR.RDCNTL` を指定した時に RDCNTL_Multiplication_factor 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.RDCNTL` を指定した時に保持した値
または、SET 処理をした値

PMA プロパティ

概要

PMA レジスタの PMA の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.PMA` を指定した時に保持されます。

Write メソッドで `REG_ADDR.PMA` を指定した時に PMA 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.PMA` を指定した時に保持した値
または、SET 処理をした値

LOOPDLY_CW プロパティ

概要

LOOPDLY_CW レジスタの LOOPDLY_CW の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.LOOPDLY_CW` を指定した時に保持されます。

Write メソッドで `REG_ADDR.LOOPDLY_CW` を指定した時に LOOPDLY_CW 値を使用して書込まれます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.LOOPDLY_CW` を指定した時に保持した値
または、SET 処理をした値

LOOPDLY_CCW プロパティ

概要

LOOPDLY_CCW レジスタの LOOPDLY_CCW の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.LOOPDLY_CCW` を指定した時に保持されます。

Write メソッドで `REG_ADDR.LOOPDLY_CCW` を指定した時に LOOPDLY_CCW 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.LOOPDLY_CCW` を指定した時に保持した値
または、SET 処理をした値

CWSCOUNT プロパティ

概要

CWSCOUNT レジスタの CWSCOUNT の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.CWSCOUNT` を指定した時に保持されます。

Write メソッドで `REG_ADDR.CWSCOUNT` を指定した時に CWSCOUNT 値を使用して書込ま
す。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.CWSCOUNT` を指定した時に保持した値
または、SET 処理をした値

CCWSCOUNT プロパティ

概要

CCWSCOUNT レジスタの CCWSCOUNT の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.CCWSCOUNT` を指定した時に保持されます。

Write メソッドで `REG_ADDR.CCWSCOUNT` を指定した時に CCWSCOUNT 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.CCWSCOUNT` を指定した時に保持した値
または、SET 処理をした値

CWPWL_PRESCALER プロパティ

概要

CWPWL、CWPWH レジスタの PRESCALER の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.CWPW` を指定した時に保持されます。

Write メソッドで `REG_ADDR.CWPW` を指定した時に CWPWL_PRESCALER 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.CWPW` を指定した時に保持した値
または、SET 処理をした値

CWPWL_STEP_PULSE_WIDTH プロパティ

概要

CWPWL、CWPWH レジスタの STEP_PULSE_WIDTH の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.CWPW` を指定した時に保持されます。

Write メソッドで `REG_ADDR.CWPW` を指定した時に CWPWL_STEP_PULSE_WIDTH 値を使用して書込ます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.CWPW` を指定した時に保持した値
または、SET 処理をした値

CCWPWL_PRESCALER プロパティ

概要

CWPWL、CWPWH レジスタの PRESCALER の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.CCWPW` を指定した時に保持されます。

Write メソッドで `REG_ADDR.CCWPW` を指定した時に `CCWPWL_PRESCALER` 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.CCWPW` を指定した時に保持した値
または、SET 処理をした値

CCWPWL_PRESCALER プロパティ

概要

CWPWL、CWPWH レジスタの STEP_PULSE_WIDTH の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.CCWPW` を指定した時に保持されます。

Write メソッドで `REG_ADDR.CCWPW` を指定した時に `CCWPWL_STEP_PULSE_WIDTH` 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.CCWPW` を指定した時に保持した値
または、SET 処理をした値

MCNTL プロパティ

概要

MCNTL レジスタの MCNTL の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.MCNTL` を指定した時に保持されます。

Write メソッドで `REG_ADDR.MCNTL` を指定した時に MCNTL 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.MCNTL` を指定した時に保持した値
または、SET 処理をした値

SUBADR1 プロパティ

概要

SUBADR1 レジスタの SUBADR1 の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.SUBADR1` を指定した時に保持されます。

Write メソッドで `REG_ADDR.SUBADR1` を指定した時に SUBADR1 値を使用して書込ます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.SUBADR1` を指定した時に保持した値
または、SET 処理をした値

SUBADR2 プロパティ

概要

SUBADR2 レジスタの SUBADR2 の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.SUBADR2` を指定した時に保持されます。

Write メソッドで `REG_ADDR.SUBADR2` を指定した時に SUBADR2 値を使用して書込ます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.SUBADR2` を指定した時に保持した値
または、SET 処理をした値

SUBADR3 プロパティ

概要

SUBADR3 レジスタの SUBADR3 の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.SUBADR3` を指定した時に保持されます。

Write メソッドで `REG_ADDR.SUBADR3` を指定した時に SUBADR3 値を使用して書込ます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.SUBADR3` を指定した時に保持した値
または、SET 処理をした値

ALLCALLADR プロパティ

概要

ALLCALLADR レジスタの ALLCALLADR の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.ALLCALLADR` を指定した時に保持されます。

Write メソッドで `REG_ADDR.ALLCALLADR` を指定した時に ALLCALLADR 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.ALLCALLADR` を指定した時に保持した値
または、SET 処理をした値

STEPCOUNT プロパティ

概要

STEPCOUNT0、STEPCOUNT1、STEPCOUNT2、STEPCOUNT3 レジスタの値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.STEPCOUNT` を指定した時に保持されます。

Write メソッドで `REG_ADDR.STEPCOUNT` を指定した時に STEPCOUNT 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.STEPCOUNT` を指定した時に保持した値
または、SET 処理をした値

ReadBuff プロパティ

概要

レジスタの値を取得した時点の変換しない値です。

処理内容

Read メソッド時に保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッド時に保持した値
または、SET 処理をした値

Init メソッド

概要

I2C バスコントローラに指定された接続設定を使って非同期的に接続します。

引数

| 引数名 | 型 | 説明 |
|---------|------|-----------|
| Address | byte | I2C のアドレス |

戻り値

なし

Read メソッド

概要

最新の変換結果を読み込みます。

引数

| 引数名 | 型 | 説明 |
|----------|------|----|
| REG_ADDR | byte | |

戻り値

Status と同じ

1: 正常

1 以外: 異常

読込処理

| REG_ADDR 値 | 読込プロパティ |
|------------|--|
| MODE | MODE |
| WDTOI | WDTOI |
| WDCNTL | WDMOD WDEN |
| IO_CFG | P0_CFG P1_CFG P2_CFG P3_CFG |
| INTMODE | P0_INTMODE P1_INTMODE P2_INTMODE P3_INTMODE P0P1_FILTER |
| MSK | P0_MSK P1_MSK P2_MSK P3_MSK MOTOR_MSK |
| INTSTAT | INTP0_FLAG INTP1_FLAG INTP2_FLAG INTP3_FLAG MOTOR_STOP_FLAG WDINT_WATCHDOG_FLAG |

| REG_ADDR 値 | 読込プロパティ |
|-------------|--|
| IP | IP |
| INT_MTR_ACT | MOTOR_CONTROL INT_AUTO INT_MOTOR |
| EXTRASTEPS0 | EXTRASTEPS0 |
| EXTRASTEPS1 | EXTRASTEPS1 |
| OP_CFG_PHS | PH_CNTL OUT_CNTL OP_LEVELS |
| OP_STAT_TO | Motor_stop_timer CCW_stop CW_stop |
| RUCNTL | RUCNTL_enable RUCNTL_change RUCNTL_Multiplication_factor |
| RDCNTL | RDCNTL_enable RDCNTL_change RDCNTL_Multiplication_factor |
| PMA | PMA |
| LOOPDLY_CW | LOOPDLY_CW |
| LOOPDLY_CCW | LOOPDLY_CCW |
| CWSCOUNT | CWSCOUNT |
| CCWSCOUNT | CCWSCOUNT |

| REG_ADDR 値 | 読込プロパティ |
|------------|---|
| CWPW | CWPWL_STEP_PULSE_WIDTH CWPWL_PRESCALER |
| CCWPW | CCWPWL_STEP_PULSE_WIDTH CCWPWL_PRESCALER |
| MCNTL | MCNTL |
| SUBADR1 | SUBADR1 |
| SUBADR2 | SUBADR2 |
| SUBADR3 | SUBADR3 |
| ALLCALLADR | ALLCALLADR |
| STEPCOUNT | STEPCOUNT |

Write メソッド

概要

引数 REG_ADDR に関するプロパティを編集し書き込みます。

引数

| 引数名 | 型 | 説明 |
|----------|------|----|
| REG_ADDR | byte | |

戻り値

Status

1: 正常

1 以外: 異常

書込処理

| REG_ADDR 値 | 書込プロパティ |
|------------|---|
| MODE | MODE |
| WDTOI | WDTOI |
| WDCNTL | WDMOD WDEN |
| IO_CFG | P0_CFG P1_CFG P2_CFG P3_CFG |
| INTMODE | P0_INTMODE P1_INTMODE P2_INTMODE P3_INTMODE P0P1_FILTER |
| MSK | P0_MSK P1_MSK P2_MSK P3_MSK MOTOR_MSK |

| REG_ADDR 値 | 書込プロパティ |
|-------------|--|
| INT_MTR_ACT | MOTOR_CONTROL INT_AUTO INT_MOTOR |
| EXTRASTEPS0 | EXTRASTEPS0 |
| EXTRASTEPS1 | EXTRASTEPS1 |
| OP_CFG_PHS | PH_CNTL OUT_CNTL OP_LEVELS |
| OP_STAT_TO | Motor_stop_timer CCW_stop CW_stop |
| RUCNTL | RUCNTL_enable RUCNTL_change RUCNTL_Multiplication_factor |
| RDCNTL | RDCNTL_enable RDCNTL_change RDCNTL_Multiplication_factor |
| PMA | PMA |
| LOOPDLY_CW | LOOPDLY_CW |
| LOOPDLY_CCW | LOOPDLY_CCW |
| CWSCOUNT | CWSCOUNT |
| CCWSCOUNT | CCWSCOUNT |

| REG_ADDR 値 | 書込プロパティ |
|------------|---|
| CWPW | CWPWL_STEP_PULSE_WIDTH CWPWL_PRESCALER |
| CCWPW | CCWPWL_STEP_PULSE_WIDTH CCWPWL_PRESCALER |
| MCNTL | MCNTL |
| SUBADR1 | SUBADR1 |
| SUBADR2 | SUBADR2 |
| SUBADR3 | SUBADR3 |
| ALLCALLADR | ALLCALLADR |

サンプルプログラム

```
Imports I2CLibrary
Public NotInheritable Class PCA9629APWPage
    Inherits Page

    Private Const PCA9629APW_Address As Byte = &H22
    Private PCA9629APW As PCA9629APW = New PCA9629APW()
    Private WithEvents RefreshTimer As DispatcherTimer = New DispatcherTimer()

    Public Sub New()
        InitializeComponent()
        Call Init()
    End Sub

    Private Async Sub Init()
        Await PCA9629APW.Init(PCA9629APW_Address)
    End Sub

    Private Sub btnStart_Click(sender As Object, e As RoutedEventArgs)
        RefreshTimer.Interval = TimeSpan.FromMilliseconds(100)

        Call SetConfig()

        'MCNTL書込
        PCA9629APW.MCNTL = &H80
        PCA9629APW.Write(PCA9629APW.REG_ADDR.MCNTL)

        RefreshTimer.Start()
    End Sub

    Private Sub Measurement(sender As Object, e As Object) Handles RefreshTimer.Tick
        RefreshTimer.Stop()

        PCA9629APW.Read(PCA9629APW.REG_ADDR.STEPCOUNT)
        Debug.WriteLine("STEPCOUNT {0:##0}", PCA9629APW.STEPCOUNT)

        PCA9629APW.Read(PCA9629APW.REG_ADDR.IP)
        Debug.WriteLine("IP:{0:D}", PCA9629APW.IP)

        PCA9629APW.Read(PCA9629APW.REG_ADDR.INTSTAT)
        Debug.WriteLine("INTP0_FLAG:{0}", PCA9629APW.INTP0_FLAG)
        Debug.WriteLine("INTP1_FLAG:{0}", PCA9629APW.INTP1_FLAG)
        Debug.WriteLine("INTP2_FLAG:{0}", PCA9629APW.INTP2_FLAG)
        Debug.WriteLine("MOTOR_STOP_FLAG:{0}", PCA9629APW.MOTOR_STOP_FLAG)
        Debug.WriteLine("WDINT_WATCHDOG_FLAG:{0}", PCA9629APW.WDINT_WATCHDOG_FLAG)

        RefreshTimer.Start()
    End Sub

    Private Sub btnStop_Click(sender As Object, e As RoutedEventArgs)
        RefreshTimer.Stop()
    End Sub
End Class
```

End Sub

Private Sub SetConfig()

'MODE書込

PCA9629APW.MODE = &H0

PCA9629APW.Write(PCA9629APW.REG_ADDR.MODE)

'OP_CFG_PHS書込

PCA9629APW.PH_CNTRL = 1 'two-phase

PCA9629APW.OUT_CNTRL = True

PCA9629APW.OP_LEVELS = &H0

PCA9629APW.Write(PCA9629APW.REG_ADDR.OP_CFG_PHS)

'CWPW書込

PCA9629APW.CWPWL_PRESCALER = PCA9629APW.Prescaler_range.MicroSec3

PCA9629APW.CWPWL_STEP_PULSE_WIDTH = 1024

PCA9629APW.Write(PCA9629APW.REG_ADDR.CWPW)

'CWPW書込

PCA9629APW.CCWPWL_PRESCALER = PCA9629APW.Prescaler_range.MicroSec3

PCA9629APW.CCWPWL_STEP_PULSE_WIDTH = &H800

PCA9629APW.Write(PCA9629APW.REG_ADDR.CCWPW)

'CWSCOUNT書込

PCA9629APW.CWSCOUNT = 512

PCA9629APW.Write(PCA9629APW.REG_ADDR.CWSCOUNT)

'CCWSCOUNT書込

PCA9629APW.CCWSCOUNT = &H80

PCA9629APW.Write(PCA9629APW.REG_ADDR.CCWSCOUNT)

'RUCNTL書込

PCA9629APW.RUCNTL_enable = True

PCA9629APW.RUCNTL_change = False

PCA9629APW.RUCNTL_Multiplication_factor = 4

PCA9629APW.Write(PCA9629APW.REG_ADDR.RUCNTL)

'RDCTNL書込

PCA9629APW.RDCNTL_enable = True

PCA9629APW.RDCNTL_change = False

PCA9629APW.RDCNTL_Multiplication_factor = 4

PCA9629APW.Write(PCA9629APW.REG_ADDR.RDCNTL)

'PMA書込

Debug.WriteLine("PMA書込")

PCA9629APW.PMA = &H2

PCA9629APW.Write(PCA9629APW.REG_ADDR.PMA)

'IO_CFG

PCA9629APW.P0_CFG = True

PCA9629APW.P1_CFG = True

```
PCA9629APW.P2_CFG = False
PCA9629APW.P3_CFG = False
PCA9629APW.Write(PCA9629APW.REG_ADDR.IO_CFG)
```

```
' INT_MTR_ACT
```

```
PCA9629APW.INT_MOTOR = &H3
PCA9629APW.INT_AUTO = &H0
PCA9629APW.MOTOR_CONTROL = True
PCA9629APW.Write(PCA9629APW.REG_ADDR.INT_MTR_ACT)
```

```
' INTMODE[10]
```

```
PCA9629APW.P0_INTMODE = False
PCA9629APW.P1_INTMODE = False
PCA9629APW.P2_INTMODE = False
PCA9629APW.P3_INTMODE = False
PCA9629APW.POP1_FILTER = &H1
PCA9629APW.Write(PCA9629APW.REG_ADDR.INTMODE)
```

```
' MSK 割り込みモーターストップ
```

```
PCA9629APW.P0_MSK = False
PCA9629APW.P1_MSK = False
PCA9629APW.P2_MSK = False
PCA9629APW.P3_MSK = False
PCA9629APW.MOTOR_MSK = False
PCA9629APW.Write(PCA9629APW.REG_ADDR.MSK)
```

```
End Sub
```

```
Private Sub btnBack_Click(sender As Object, e As RoutedEventArgs)
```

```
    RefreshTimer.Stop()
    PCA9629APW.close()
    App.rootFrame.Navigate(GetType(MenuPage))
```

```
End Sub
```

```
End Class
```

3-8.ジェスチャーセンサー PAJ7620U2

変更履歴

2016年9月1日 第一版 新規作成

1. 概要

エリアセンサー(PAJ7620U2)とI2C接続するための機能を提供します。

2. 機能一覧

プロパティ

| 機能名 | 機能説明 |
|------------------------|--|
| REGISTER_BANK | Register Bank を設定します。 |
| Suspend | REGISTER_BANK が 0 の場合、サスペンドを設定します。 |
| Enable | REGISTER_BANK が 1 の場合、PAJ7620U2 の Enabled/Disable を取得または設定します。 |
| Counter_Clockwise_Mask | REGISTER_BANK が 0 の場合、反時計回りのジェスチャーのマスクを取得または設定します。 |
| Clockwise_Mask | REGISTER_BANK が 0 の場合、時計回りのジェスチャーのマスクを取得または設定します。 |
| Backward_Mask | REGISTER_BANK が 0 の場合、後退のジェスチャーのマスクを取得または設定します。 |
| Forward_Mask | REGISTER_BANK が 0 の場合、前進のジェスチャーのマスクを取得または設定します。 |
| Right_Mask | REGISTER_BANK が 0 の場合、右ジェスチャーのマスクを取得または設定します。 |
| Left_Mask | REGISTER_BANK が 0 の場合、左ジェスチャーのマスクを取得または設定します。 |
| Down_Mask | REGISTER_BANK が 0 の場合、下ジェスチャーのマスクを取得または設定します。 |
| Up_Mask | REGISTER_BANK が 0 の場合、上ジェスチャーのマスクを取得または設定します。 |
| Wave_Mask | REGISTER_BANK が 0 の場合、ウェーブジェスチャーのマスクを取得または設定します。 |

| 機能名 | 機能説明 |
|-------------------|---|
| Counter_Clockwise | REGISTER_BANK が 0 の場合、反時計回りのジェスチャーを取得します。 |
| Clockwise | REGISTER_BANK が 0 の場合、時計回りのジェスチャーを取得します。 |
| Backward | REGISTER_BANK が 0 の場合、後退のジェスチャーを取得します。 |
| Forward | REGISTER_BANK が 0 の場合、前進のジェスチャーを取得します。 |
| Right | REGISTER_BANK が 0 の場合、右ジェスチャーを取得します。 |
| Left | REGISTER_BANK が 0 の場合、左ジェスチャーを取得します。 |
| Down | REGISTER_BANK が 0 の場合、下ジェスチャーを取得します。 |
| Up | REGISTER_BANK が 0 の場合、上ジェスチャーを取得します。 |
| Wave | REGISTER_BANK が 0 の場合、ウェーブジェスチャーを取得します。 |
| State | REGISTER_BANK が 0 の場合、状態を取得します。 |
| IDLE_S1_Step | REGISTER_BANK が 1 の場合、IDLE_S1_Step を取得または設定します。 |
| IDLE_S2_Step | REGISTER_BANK が 1 の場合、IDLE_S2_Step を取得または設定します。 |
| OPTOS1_Step | REGISTER_BANK が 1 の場合、OPTOS1_Step を取得または設定します。 |
| S1TOS2_Step | REGISTER_BANK が 1 の場合、S1TOS2_Step を取得または設定します。 |
| ObjectBrightness | REGISTER_BANK が 1 の場合、ObjectBrightness を取得または設定します。 |
| ObjectSize | REGISTER_BANK が 1 の場合、ObjectSize を取得または設定します。 |
| PSGain | REGISTER_BANK が 1 の場合、PSGain を取得または設定します。 |
| Proximity_Mask | REGISTER_BANK が 0 の場合、近接検出のマスクを取得または設定します。 |

| 機能名 | 機能説明 |
|---------------|--|
| Proximity | REGISTER_BANK が 0 の場合、近接検出を取得します。 |
| PS_High_Thd | REGISTER_BANK が 1 の場合、PS_High_Thd を取得または設定します。 |
| PS_Low_Thd | REGISTER_BANK が 1 の場合、PS_Low_Thd を取得または設定します。 |
| PS_Approach | REGISTER_BANK が 1 の場合、PS_Approach を取得します。 |
| PS_8bits_Data | REGISTER_BANK が 1 の場合、PS_8bits_Data を取得します。 |

メソッド

| 機能名 | 機能説明 |
|-------|---------------------------------------|
| Init | I2C バスコントローラに指定された接続設定を使って非同期的に接続します。 |
| Read | 指定レジスタ読込 |
| Write | 指定レジスタ書込 |

定数

| 機能名 | 機能説明 | 定数値 |
|--|------------------------------------|------|
| REG_ADDR.REGISTER_BANK | レジスタバンク | 0xEF |
| REG_ADDR.Suspend | レジスタバンク 0、サスペンド | 0x03 |
| REG_ADDR. GESTURE_DETECTION_INTERRUPT_FLAG_MASK | レジスタバンク 0、ジェスチャー検出割り込み フラグマスク | 0x41 |
| REG_ADDR. GESTURE_DETECTION_INTERRUPT_FLAG | レジスタバンク 0、ジェスチャー検出割り込み フラグ | 0x43 |
| REG_ADDR.State | レジスタバンク 0、ジェスチャー検出のための 状態インジケータ | 0x45 |
| REG_ADDR.PS_High_Thd | レジスタバンク 0、PS ヒステリシス高閾値 | 0x69 |
| REG_ADDR.PS_Low_Thd | レジスタバンク 0、PS ヒステリシス低閾値 | 0x6A |
| REG_ADDR.PS_Approach | レジスタバンク 0、PS アプローチ状態 | 0x6B |
| REG_ADDR.PS_8bits_Data | レジスタバンク 0、8bit データ | 0x6C |
| REG_ADDR.ObjectBrightness | レジスタバンク 0、被写体輝度 | 0xB0 |
| REG_ADDR.ObjectSize | レジスタバンク 0、オブジェクトサイズ | 0xB1 |
| REG_ADDR.PS_Gain | レジスタバンク 1、PS ゲイン設定 | 0x44 |
| REG_ADDR.IDLE_S1_Step | レジスタバンク 1、S1、レスポンスファクタ | 0x67 |
| REG_ADDR.IDLE_S2_Step | レジスタバンク 1、S2、レスポンスファクタ | 0x69 |
| REG_ADDR.OPTOS1_Step | レジスタバンク 1、OPToS1 ステップ時間 | 0x6B |
| REG_ADDR.S1TOS2_Step | レジスタバンク 1、S1toS2 ステップ時間 | 0x6D |
| REG_ADDR.Enable | レジスタバンク 1、PAJ7620U2 有効/無効 | 0x72 |

REGISTER_BANK プロパティ

概要

Register Bank を設定します。

処理内容

Write メソッドで `REG_ADDR.REGISTER_BANK` を指定した時に `REGISTER_BANK` 値を使用して書込みます。

0: RegisterBank 0

1: RegisterBank 1

SET 処理

value 値をセットする。

GET 処理

SET 処理をした値

Suspend プロパティ

概要

REGISTER_BANK が 0 の場合、サスペンドを設定します。

処理内容

Write メソッドで `REG_ADDR.Suspend` を指定した時に Suspend 値を使用して書込みます。

1: サスペンド状態(デフォルト)

SET 処理

value 値をセットする。

GET 処理

SET 処理をした値

Enable プロパティ

概要

REGISTER_BANK が 1 の場合、PAJ7620U2 の Enabled/Disable を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Enable` を指定した時に保持されます。

Write メソッドで `REG_ADDR.Enable` を指定した時に Enable 値を使用して書込みます。

1: PAJ7620U2 有効

0: PAJ7620U2 無効

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Enable` を指定した時に保持した値

または、SET 処理をした値

Counter_Clockwise_Mask プロパティ

概要

REGISTER_BANK が 0 の場合、反時計回りのジェスチャーのマスクを取得または設定します。

処理内容

Read メソッドで `REG_ADDR.GESTURE_DETECTION_INTERRUPT_FLAG_MASK` を指定した時に保持されます。

Write メソッドで `REG_ADDR.GESTURE_DETECTION_INTERRUPT_FLAG_MASK` を指定した時に Counter_Clockwise_Mask 値を使用して書込みます。

1: 割り込みを生成します

0: 割り込みを生成しません

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.GESTURE_DETECTION_INTERRUPT_FLAG_MASK` を指定した時に保持した値

または、SET 処理をした値

Clockwise_Mask プロパティ

概要

REGISTER_BANK が 0 の場合、時計回りのジェスチャーのマスクを取得または設定します。

処理内容

Read メソッドで `REG_ADDR.GESTURE_DETECTION_INTERRUPT_FLAG_MASK` を指定した時に保持されます。

Write メソッドで `REG_ADDR.GESTURE_DETECTION_INTERRUPT_FLAG_MASK` を指定した時に Clockwise_Mask 値を使用して書込みます。

1: 割り込みを生成します

0: 割り込みを生成しません

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.GESTURE_DETECTION_INTERRUPT_FLAG_MASK` を指定した時に保持した値

または、SET 処理をした値

Backward_Mask プロパティ

概要

REGISTER_BANK が 0 の場合、後退のジェスチャーのマスクを取得または設定します。

処理内容

Read メソッドで `REG_ADDR.GESTURE_DETECTION_INTERRUPT_FLAG_MASK` を指定した時に保持されます。

Write メソッドで `REG_ADDR.GESTURE_DETECTION_INTERRUPT_FLAG_MASK` を指定した時に Backward_Mask 値を使用して書込みます。

1: 割り込みを生成します

0: 割り込みを生成しません

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.GESTURE_DETECTION_INTERRUPT_FLAG_MASK` を指定した時に保持した値

または、SET 処理をした値

Forward_Mask プロパティ

概要

REGISTER_BANK が 0 の場合、前進のジェスチャーのマスクを取得または設定します。

処理内容

Read メソッドで `REG_ADDR.GESTURE_DETECTION_INTERRUPT_FLAG_MASK` を指定した時に保持されます。

Write メソッドで `REG_ADDR.GESTURE_DETECTION_INTERRUPT_FLAG_MASK` を指定した時に Forward_Mask 値を使用して書込みます。

1: 割り込みを生成します

0: 割り込みを生成しません

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.GESTURE_DETECTION_INTERRUPT_FLAG_MASK` を指定した時に保持した値

または、SET 処理をした値

Right_Mask プロパティ

概要

REGISTER_BANK が 0 の場合、右ジェスチャーのマスクを取得または設定します。

処理内容

Read メソッドで `REG_ADDR.GESTURE_DETECTION_INTERRUPT_FLAG_MASK` を指定した時に保持されます。

Write メソッドで `REG_ADDR.GESTURE_DETECTION_INTERRUPT_FLAG_MASK` を指定した時に Right_Mask 値を使用して書込みます。

1: 割り込みを生成します

0: 割り込みを生成しません

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.GESTURE_DETECTION_INTERRUPT_FLAG_MASK` を指定した時に保持した値

または、SET 処理をした値

Left_Mask プロパティ

概要

REGISTER_BANK が 0 の場合、左ジェスチャーのマスクを取得または設定します。

処理内容

Read メソッドで `REG_ADDR.GESTURE_DETECTION_INTERRUPT_FLAG_MASK` を指定した時に保持されます。

Write メソッドで `REG_ADDR.GESTURE_DETECTION_INTERRUPT_FLAG_MASK` を指定した時に Left_Mask 値を使用して書込みます。

1: 割り込みを生成します

0: 割り込みを生成しません

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.GESTURE_DETECTION_INTERRUPT_FLAG_MASK` を指定した時に保持した値

または、SET 処理をした値

Down_Mask プロパティ

概要

REGISTER_BANK が 0 の場合、下ジェスチャーのマスクを取得または設定します。

処理内容

Read メソッドで `REG_ADDR.GESTURE_DETECTION_INTERRUPT_FLAG_MASK` を指定した時に保持されます。

Write メソッドで `REG_ADDR.GESTURE_DETECTION_INTERRUPT_FLAG_MASK` を指定した時に Down_Mask 値を使用して書込みます。

1: 割り込みを生成します

0: 割り込みを生成しません

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.GESTURE_DETECTION_INTERRUPT_FLAG_MASK` を指定した時に保持した値

または、SET 処理をした値

Up_Mask プロパティ

概要

REGISTER_BANK が 0 の場合、上ジェスチャーのマスクを取得または設定します。

処理内容

Read メソッドで `REG_ADDR.GESTURE_DETECTION_INTERRUPT_FLAG_MASK` を指定した時に保持されます。

Write メソッドで `REG_ADDR.GESTURE_DETECTION_INTERRUPT_FLAG_MASK` を指定した時に Up_Mask 値を使用して書込みます。

1: 割り込みを生成します

0: 割り込みを生成しません

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.GESTURE_DETECTION_INTERRUPT_FLAG_MASK` を指定した時に保持した値

または、SET 処理をした値

Wave_Mask プロパティ

概要

REGISTER_BANK が 0 の場合、ウェーブジェスチャーのマスクを取得または設定します。

処理内容

Read メソッドで `REG_ADDR.GESTURE_DETECTION_INTERRUPT_FLAG_MASK` を指定した時に保持されます。

Write メソッドで `REG_ADDR.GESTURE_DETECTION_INTERRUPT_FLAG_MASK` を指定した時に Wave_Mask 値を使用して書込みます。

1: 割り込みを生成します

0: 割り込みを生成しません

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.GESTURE_DETECTION_INTERRUPT_FLAG_MASK` を指定した時に保持した値

または、SET 処理をした値

Counter_Clockwise プロパティ

概要

REGISTER_BANK が 0 の場合、反時計回りのジェスチャーを取得します。

処理内容

Read メソッドで `REG_ADDR.GESTURE_DETECTION_INTERRUPT_FLAG` を指定した時に保持されます。

1: ジェスチャー検出

0: ジェスチャー未検出

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.GESTURE_DETECTION_INTERRUPT_FLAG` を指定した時に保持した値

または、SET 処理をした値

Clockwise プロパティ

概要

REGISTER_BANK が 0 の場合、時計回りのジェスチャーを取得します。

処理内容

Read メソッドで `REG_ADDR.GESTURE_DETECTION_INTERRUPT_FLAG` を指定した時に保持されます。

1: ジェスチャー検出

0: ジェスチャー未検出

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.GESTURE_DETECTION_INTERRUPT_FLAG` を指定した時に保持した値

または、SET 処理をした値

Backward プロパティ

概要

REGISTER_BANK が 0 の場合、後退のジェスチャーを取得します。

処理内容

Read メソッドで `REG_ADDR.GESTURE_DETECTION_INTERRUPT_FLAG` を指定した時に保持されます。

1: ジェスチャー検出

0: ジェスチャー未検出

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.GESTURE_DETECTION_INTERRUPT_FLAG` を指定した時に保持した値

または、SET 処理をした値

Forward プロパティ

概要

REGISTER_BANK が 0 の場合、前進のジェスチャーを取得します。

処理内容

Read メソッドで `REG_ADDR.GESTURE_DETECTION_INTERRUPT_FLAG` を指定した時に保持されます。

1: ジェスチャー検出

0: ジェスチャー未検出

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.GESTURE_DETECTION_INTERRUPT_FLAG` を指定した時に保持した値

または、SET 処理をした値

Right プロパティ

概要

REGISTER_BANK が 0 の場合、右ジェスチャーを取得します。

処理内容

Read メソッドで `REG_ADDR.GESTURE_DETECTION_INTERRUPT_FLAG` を指定した時に保持されます。

1: ジェスチャー検出

0: ジェスチャー未検出

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.GESTURE_DETECTION_INTERRUPT_FLAG` を指定した時に保持した値

または、SET 処理をした値

Left プロパティ

概要

REGISTER_BANK が 0 の場合、左ジェスチャーを取得します。

処理内容

Read メソッドで `REG_ADDR.GESTURE_DETECTION_INTERRUPT_FLAG` を指定した時に保持されます。

1: ジェスチャー検出

0: ジェスチャー未検出

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.GESTURE_DETECTION_INTERRUPT_FLAG` を指定した時に保持した値

または、SET 処理をした値

Down プロパティ

概要

REGISTER_BANK が 0 の場合、下ジェスチャーを取得します。

処理内容

Read メソッドで `REG_ADDR.GESTURE_DETECTION_INTERRUPT_FLAG` を指定した時に保持されます。

1: ジェスチャー検出

0: ジェスチャー未検出

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.GESTURE_DETECTION_INTERRUPT_FLAG` を指定した時に保持した値

または、SET 処理をした値

Up プロパティ

概要

REGISTER_BANK が 0 の場合、上ジェスチャーを取得します。

処理内容

Read メソッドで `REG_ADDR.GESTURE_DETECTION_INTERRUPT_FLAG` を指定した時に保持されます。

1: ジェスチャー検出

0: ジェスチャー未検出

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.GESTURE_DETECTION_INTERRUPT_FLAG` を指定した時に保持した値

または、SET 処理をした値

Wave プロパティ

概要

REGISTER_BANK が 0 の場合、ウェーブジェスチャーを取得します。

処理内容

Read メソッドで `REG_ADDR.GESTURE_DETECTION_INTERRUPT_FLAG` を指定した時に保持されます。

1: ジェスチャー検出

0: ジェスチャー未検出

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.GESTURE_DETECTION_INTERRUPT_FLAG` を指定した時に保持した値

または、SET 処理をした値

State プロパティ

概要

REGISTER_BANK が 0 の場合、状態を取得します。

処理内容

Read メソッドで `REG_ADDR.State` を指定した時に保持されます。

0: 動作状態

1: スタンバイ 1 状態

2: スタンバイ 2 状態

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.State` を指定した時に保持した値
または、SET 処理をした値

IDLE_S1_Step プロパティ

概要

REGISTER_BANK が 1 の場合、IDLE_S1_Step を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.IDLE_S1_Step` を指定した時に保持されます。

Write メソッドで `REG_ADDR.IDLE_S1_Step` を指定した時に IDLE_S1_Step 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.IDLE_S1_Step` を指定した時に保持した値
または、SET 処理をした値

IDLE_S2_Step プロパティ

概要

REGISTER_BANK が 1 の場合、IDLE_S2_Step を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.IDLE_S2_Step` を指定した時に保持されます。

Write メソッドで `REG_ADDR.IDLE_S2_Step` を指定した時に IDLE_S2_Step 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.IDLE_S2_Step` を指定した時に保持した値
または、SET 処理をした値

OPTOS1_Step プロパティ

概要

REGISTER_BANK が 1 の場合、OPTOS1_Step を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.OPTOS1_Step` を指定した時に保持されます。

Write メソッドで `REG_ADDR.OPTOS1_Step` を指定した時に `OPTOS1_Step` 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.OPTOS1_Step` を指定した時に保持した値
または、SET 処理をした値

S1TOS2_Step プロパティ

概要

REGISTER_BANK が 1 の場合、S1TOS2_Step を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.S1TOS2_Step` を指定した時に保持されます。

Write メソッドで `REG_ADDR.S1TOS2_Step` を指定した時に S1TOS2_Step 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.S1TOS2_Step` を指定した時に保持した値
または、SET 処理をした値

ObjectBrightness プロパティ

概要

REGISTER_BANK が 1 の場合、ObjectBrightness を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.ObjectBrightness` を指定した時に保持されます。

Write メソッドで `REG_ADDR.ObjectBrightness` を指定した時に ObjectBrightness 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.ObjectBrightness` を指定した時に保持した値
または、SET 処理をした値

ObjectSize プロパティ

概要

REGISTER_BANK が 1 の場合、ObjectSize を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.ObjectSize` を指定した時に保持されます。

Write メソッドで `REG_ADDR.ObjectSize` を指定した時に ObjectSize 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.ObjectSize` を指定した時に保持した値
または、SET 処理をした値

PSGain プロパティ

概要

REGISTER_BANK が 1 の場合、PSGain を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.PS_Gain` を指定した時に保持されます。

Write メソッドで `REG_ADDR.PS_Gain` を指定した時に PSGain 値を使用して書込みます。

0:1 倍

1:2 倍

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.PS_Gain` を指定した時に保持した値

または、SET 処理をした値

Proximity_Mask プロパティ

概要

REGISTER_BANK が 0 の場合、近接検出のマスクを取得または設定します。

処理内容

Read メソッドで `REG_ADDR.GESTURE_DETECTION_INTERRUPT_FLAG_MASK` を指定した時に保持されます。

Write メソッドで `REG_ADDR.GESTURE_DETECTION_INTERRUPT_FLAG_MASK` を指定した時に Proximity_Mask 値を使用して書込みます。

1: 割り込みを生成します

0: 割り込みを生成しません

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.GESTURE_DETECTION_INTERRUPT_FLAG_MASK` を指定した時に保持した値

または、SET 処理をした値

Proximity プロパティ

概要

REGISTER_BANK が 0 の場合、近接検出を取得します。

処理内容

Read メソッドで `REG_ADDR.GESTURE_DETECTION_INTERRUPT_FLAG` を指定した時に保持されます。

1: ジェスチャー検出

0: ジェスチャー未検出

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.GESTURE_DETECTION_INTERRUPT_FLAG` を指定した時に保持した値

または、SET 処理をした値

PS_High_Thd プロパティ

概要

REGISTER_BANK が 1 の場合、PS_High_Thd を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.PS_High_Thd` を指定した時に保持されます。

Write メソッドで `REG_ADDR.PS_High_Thd` を指定した時に PS_High_Thd 値を使用して書込まず。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.PS_High_Thd` を指定した時に保持した値
または、SET 処理をした値

PS_Low_Thd プロパティ

概要

REGISTER_BANK が 1 の場合、PS_Low_Thd を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.PS_Low_Thd` を指定した時に保持されます。

Write メソッドで `REG_ADDR.PS_Low_Thd` を指定した時に PS_Low_Thd 値を使用して書込まず。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.PS_Low_Thd` を指定した時に保持した値
または、SET 処理をした値

PS_Approach プロパティ

概要

REGISTER_BANK が 1 の場合、PS_Approach を取得します。

処理内容

Read メソッドで `REG_ADDR.PS_Approach` を指定した時に保持されます。

PS アプローチ状態

アプローチ=1、(8ビット PS データ) >= PS 高しきい値)

アプローチではない=0(8ビット PS データ <= PS 低閾値)

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.PS_Approach` を指定した時に保持した値

または、SET 処理をした値

PS_8bits_Data プロパティ

概要

REGISTER_BANK が 1 の場合、PS_Approach を取得します。

処理内容

Read メソッドで `REG_ADDR.PS_8bits_Data` を指定した時に保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.PS_8bits_Data` を指定した時に保持した値
または、SET 処理をした値

Init メソッド

概要

I2C バスコントローラに指定された接続設定を使って非同期的に接続します。

引数

| 引数名 | 型 | 説明 |
|---------|------|-----------|
| Address | byte | I2C のアドレス |

戻り値

なし

Read メソッド

概要

最新の変換結果を読みます。

引数

| 引数名 | 型 | 説明 |
|----------|------|--|
| REG_ADDR | byte | REG_ADDR_Suspend(0x03) ~ REG_ADDR_REGISTER_BANK (0xEF) |

戻り値

Status と同じ

1: 正常

1 以外: 異常

読込処理

| REG_ADDR 値 | 読込プロパティ |
|---------------------------------------|---|
| GESTURE_DETECTION_INTERRUPT_FLAG_MASK | Right_Mask Left_Mask Down_Mask Up_Mask Forward_Mask Backward_Mask Counter_Clockwise_Mask Clockwise_Mask Wave_Mask Proximity_Mask |
| GESTURE_DETECTION_INTERRUPT_FLAG | Right Left Down Up Forward Backward Counter_Clockwise Clockwise Wave Proximity |

| REG_ADDR 値 | 読込プロパティ |
|------------------|------------------|
| State | State |
| PS_High_Thd | PS_High_Thd |
| PS_Low_Thd | PS_Low_Thd |
| PS_Approach | PS_Approach |
| PS_8bits_Data | PS_8bits_Data |
| ObjectBrightness | ObjectBrightness |
| ObjectSize | ObjectSize |
| PS_Gain | PSGain |
| IDLE_S1_Step | IDLE_S1_Step |
| IDLE_S2_Step | IDLE_S2_Step |
| OPTOS1_Step | OPTOS1_Step |
| S1TOS2_Step | S1TOS2_Step |
| Enable | Enable |

Write メソッド

概要

引数 REG_ADDR に関するプロパティを編集し書き込みます。

引数

| 引数名 | 型 | 説明 |
|----------|------|--|
| REG_ADDR | byte | REG_ADDR_Suspend(0x03) ~ REG_ADDR_REGISTER_BANK (0xEF) |

戻り値

Status

1: 正常

1 以外: 異常

書込処理

| REG_ADDR 値 | 読込プロパティ |
|---------------------------------------|---|
| REGISTER_BANK | REGISTER_BANK |
| Suspend | Suspend |
| GESTURE_DETECTION_INTERRUPT_FLAG_MASK | Right_Mask Left_Mask Down_Mask Up_Mask Forward_Mask Backward_Mask Counter_Clockwise_Mask Clockwise_Mask Wave_Mask Proximity_Mask |
| PS_High_Thd | PS_High_Thd |
| PS_Low_Thd | PS_Low_Thd |
| PS_Gain | PSGain |
| IDLE_S1_Step | IDLE_S1_Step |
| IDLE_S2_Step | IDLE_S2_Step |
| OPTOS1_Step | OPTOS1_Step |
| S1TOS2_Step | S1TOS2_Step |
| Enable | Enable |
| | |

サンプルプログラム

```
Imports I2CLibrary
Public NotInheritable Class PAJ7620U2Page
    Inherits Page

    Private Const PAJ7620U2_Address As Byte = &H73
    Private PAJ7620U2 As PAJ7620U2 = New PAJ7620U2()
    Private WithEvents RefreshTimer As DispatcherTimer = New DispatcherTimer()

    Public Sub New()
        InitializeComponent()
        Call Init()
    End Sub

    Private Async Sub Init()
        Await PAJ7620U2.Init(PAJ7620U2_Address)
    End Sub

    Private Sub btnStart_Click(sender As Object, e As RoutedEventArgs)
        PAJ7620U2.REGISTER_BANK = &H0
        PAJ7620U2.Write(PAJ7620U2.REG_ADDR.REGISTER_BANK)

        PAJ7620U2.Suspend = &H0
        PAJ7620U2.Write(PAJ7620U2.REG_ADDR.Suspend)

        PAJ7620U2.REGISTER_BANK = &H1
        PAJ7620U2.Write(PAJ7620U2.REG_ADDR.REGISTER_BANK)

        PAJ7620U2.Enable = &H1
        PAJ7620U2.Write(PAJ7620U2.REG_ADDR.Enable)

        PAJ7620U2.REGISTER_BANK = &H0
        PAJ7620U2.Write(PAJ7620U2.REG_ADDR.REGISTER_BANK)

        PAJ7620U2.Up_Mask = &H1
        PAJ7620U2.Down_Mask = &H1
        PAJ7620U2.Left_Mask = &H1
        PAJ7620U2.Right_Mask = &H1
        PAJ7620U2.Forward_Mask = &H1
        PAJ7620U2.Backward_Mask = &H1
        PAJ7620U2.Counter_Clockwise_Mask = &H1
        PAJ7620U2.Clockwise_Mask = &H1
        PAJ7620U2.Wave_Mask = &H1
        PAJ7620U2.Proximity = &H0

        PAJ7620U2.Write(PAJ7620U2.REG_ADDR.GESTURE_DETECTION_INTERRUPT_FLAG_MASK)

        RefreshTimer.Interval = TimeSpan.FromMilliseconds(100)
        RefreshTimer.Start()
    End Sub

    Private Sub Measurement(sender As Object, e As Object) Handles RefreshTimer.Tick
```

```

RefreshTimer.Stop()

PAJ7620U2.REGISTER_BANK = &H0
PAJ7620U2.Write(PAJ7620U2.REG_ADDR.REGISTER_BANK)

PAJ7620U2.Read(PAJ7620U2.REG_ADDR.GESTURE_DETECTION_INTERRUPT_FLAG)
PAJ7620U2.Read(PAJ7620U2.REG_ADDR.State)

If (PAJ7620U2.Up = 1) Then
    Debug.WriteLine("Up")
End If
If (PAJ7620U2.Down = 1) Then
    Debug.WriteLine("Down")
End If
If (PAJ7620U2.Left = 1) Then
    Debug.WriteLine("Left")
End If
If (PAJ7620U2.Right = 1) Then
    Debug.WriteLine("Right")
End If
If (PAJ7620U2.Forward = 1) Then
    Debug.WriteLine("Forward")
End If
If (PAJ7620U2.Backward = 1) Then
    Debug.WriteLine("Backward")
End If
If (PAJ7620U2.Clockwise = 1) Then
    Debug.WriteLine("Clockwise")
End If
If (PAJ7620U2.Counter_Clockwise = 1) Then
    Debug.WriteLine("Counter_Clockwise")
End If
If (PAJ7620U2.Wave = 1) Then
    Debug.WriteLine("Wave")
End If

RefreshTimer.Start()
End Sub

Private Sub btnStop_Click(sender As Object, e As RoutedEventArgs)
    RefreshTimer.Stop()
End Sub

Private Sub btnBack_Click(sender As Object, e As RoutedEventArgs)
    RefreshTimer.Stop()
    PAJ7620U2.close()
    App.rootFrame.Navigate(GetType(MenuPage))
End Sub
End Class

```

3-9.液晶モジュールコントローラ PIC16F77A

変更履歴

2016年12月05日 第一版 新規作成

1. 概要

液晶モジュールコントローラ(GLOVE LCD RGB Backlight)とI2C 接続するための機能を提供します。

2. 機能一覧

プロパティ

| 機能名 | 機能説明 |
|-----|--------------------------------|
| I_D | true:右, false:左 |
| S | true:表示シフト ON, false:表示シフト OFF |
| D | true:ディスプレイ ON |
| C | true:ブロックカーソル ON |
| B | true:ブリンク ON |
| S_C | true:表示シフト, false:カーソル移動 |
| R_L | true:右, false:左 |
| DL | true:8bit, false:4bit |
| N | true:2line, false:1line |
| F | true:5x10dot, false:5x7dot |
| col | 列 |
| row | 行 |

| 機能名 | 機能説明 |
|------------|--------------|
| Data | 表示データ |
| CharMap | ユーザー定義文字 |
| location | 位置 |
| WriteDelay | 書込み文字間 Delay |
| ReadBuff | 読込データ |

メソッド

| 機能名 | 機能説明 |
|-------|---------------------------------------|
| Init | I2C バスコントローラに指定された接続設定を使って非同期的に接続します。 |
| Write | 指定レジスタ書込 |

定数

| 機能名 | 機能説明 | 定数值 |
|------------------------|---------------|-----|
| REG_ADDR.INT | データ書き込み | |
| REG_ADDR.LEARDISPLAY | ディスプレイのクリア | |
| REG_ADDR.ETURNHOME | カーソルをホーム | |
| REG_ADDR.NTRYMODESET | エントリーモードセット | |
| REG_ADDR.ISPLAYCONTROL | 表示 On/Off | |
| REG_ADDR.URSORSHIFT | カーソル/シフト設定 | |
| REG_ADDR.UNCTIONSET | 機能設定 | |
| REG_ADDR.ETCGRAMADDR | CGRAM アドレスセット | |
| REG_ADDR.ETDDRAMADDR | DDRAM アドレスセット | |

I_D プロパティ

概要

ENTRYMODESET レジスタの I_D の値を取得または設定します。

処理内容

Write メソッドで ENTRYMODESET を指定した時に I_D 値を使用して書込みます。

SET 処理

value 値をセットする。

S プロパティ

概要

ENTRYMODESET レジスタの S の値を取得または設定します。

処理内容

Write メソッドで ENTRYMODESET を指定した時に S 値を使用して書込みます。

SET 処理

value 値をセットする。

D プロパティ

概要

DISPLAYCONTROL レジスタの D の値を取得または設定します。

処理内容

Write メソッドで DISPLAYCONTROL を指定した時に D 値を使用して書込みます。

SET 処理

value 値をセットする。

C プロパティ

概要

DISPLAYCONTROL レジスタの C の値を取得または設定します。

処理内容

Write メソッドで DISPLAYCONTROL を指定した時に C 値を使用して書込みます。

SET 処理

value 値をセットする。

B プロパティ

概要

DISPLAYCONTROL レジスタの B の値を取得または設定します。

処理内容

Write メソッドで DISPLAYCONTROL を指定した時に B 値を使用して書込みます。

SET 処理

value 値をセットする。

S_C プロパティ

概要

CURSORSHIFT レジスタの S_C の値を取得または設定します。

処理内容

Write メソッドで CURSORSHIFT を指定した時に S_C 値を使用して書込みます。

SET 処理

value 値をセットする。

R_L プロパティ

概要

CURSORSHIFT レジスタの R_L の値を取得または設定します。

処理内容

Write メソッドで CURSORSHIFT を指定した時に R_L 値を使用して書込みます。

SET 処理

value 値をセットする。

R_L プロパティ

概要

FUNCTIONSET レジスタの DL の値を取得または設定します。

処理内容

Write メソッドで FUNCTIONSET を指定した時に DL 値を使用して書込みます。

SET 処理

value 値をセットする。

N プロパティ

概要

FUNCTIONSET レジスタの N の値を取得または設定します。

処理内容

Write メソッドで FUNCTIONSET を指定した時に N 値を使用して書込みます。

SET 処理

value 値をセットする。

F プロパティ

概要

FUNCTIONSET レジスタの F の値を取得または設定します。

処理内容

Write メソッドで FUNCTIONSET を指定した時に F 値を使用して書込みます。

SET 処理

value 値をセットする。

col プロパティ

概要

SETDDRAMADDR レジスタの col の値を取得または設定します。

処理内容

Write メソッドで SETDDRAMADDR を指定した時に col 値を使用して書込みます。

SET 処理

value 値をセットする。

row プロパティ

概要

SETDDRAMADDR レジスタの row の値を取得または設定します。

処理内容

Write メソッドで SETDDRAMADDR を指定した時に row 値を使用して書込みます。

SET 処理

value 値をセットする。

Data プロパティ

概要

PRINT レジスタの Data の値を取得または設定します。

処理内容

Write メソッドで PRINT を指定した時に Data 値を使用して書込みます。

SET 処理

value 値をセットする。

WriteDelay プロパティ

概要

PRINT レジスタの WriteDelay の値を取得または設定します。

処理内容

Write メソッドで PRINT を指定した時に WriteDelay 値を使用して書込みます。

SET 処理

value 値をセットする。

CharMap プロパティ

概要

SETCGRAMADDR レジスタの CharMap の値を取得または設定します。

処理内容

Write メソッドで SETCGRAMADDR を指定した時に CharMap 値を使用して書込みます。

SET 処理

value 値をセットする。

location プロパティ

概要

SETGRAMADDR レジスタの location の値を取得または設定します。

処理内容

Write メソッドで SETGRAMADDR を指定した時に location 値を使用して書込みます。

SET 処理

value 値をセットする。

Init メソッド

概要

I2C バスコントローラに指定された接続設定を使って非同期的に接続します。

引数

| 引数名 | 型 | 説明 |
|---------|------|-----------|
| Address | byte | I2C のアドレス |

戻り値

なし

Write メソッド

概要

引数 RegSelect に関するプロパティを編集し書き込みます。

引数

| 引数名 | 型 | 説明 |
|-----------|------|-----------|
| RegSelect | byte | 0x01～0x07 |

戻り値

Status

1: 正常

1 以外: 異常

書込処理

| REG_ADDR 値 | 書込パティ |
|----------------|---------------------|
| PRINT | Data WriteDelay |
| CLEARDISPLAY | |
| RETURNHOME | |
| ENTRYMODESET | I_D S |
| DISPLAYCONTROL | D C B |
| CURSORSHIFT | S_C R_L |
| FUNCTIONSET | DL N F |
| SETCGRAMADDR | location CharMap |
| SETDDRAMADDR | row col |

サンプルプログラム

```
Imports I2CLibrary
Public NotInheritable Class PIC16F77APage
    Inherits Page
    Private Const GROVE_LCD_Address As Byte = &H3E
    Private GROVE_LCD As I2CLibrary.GROVE_LCD = New I2CLibrary.GROVE_LCD()

    Public Sub New()
        InitializeComponent()
        Call Init()
    End Sub

    Private Async Sub Init()
        Await GROVE_LCD.Init(GROVE_LCD_Address)
    End Sub

    Private Async Sub btnLCD_Click(sender As Object, e As RoutedEventArgs)
        ' 1.機能設定
        GROVE_LCD.DL = False    ' DB=4Bit
        GROVE_LCD.N = True     ' 1/16 duty
        GROVE_LCD.F = False    ' 2行
        Await GROVE_LCD.Write(I2CLibrary.GROVE_LCD.REG_ADDR.FUNCTIONSET)
        ' 2.表示関係
        GROVE_LCD.D = True     ' 表示On
        GROVE_LCD.C = True     ' カーソルOff
        GROVE_LCD.B = False    ' カーソル点減なし
        Await GROVE_LCD.Write(I2CLibrary.GROVE_LCD.REG_ADDR.DISPLAYCONTROL)
        ' 3.ディスプレイのクリア
        Await GROVE_LCD.Write(I2CLibrary.GROVE_LCD.REG_ADDR.CLEARDISPLAY)
        ' 3.カーソル移動
        GROVE_LCD.I.D = True   ' ACを+1
        GROVE_LCD.S = False    ' カーソルシフト
        Await GROVE_LCD.Write(I2CLibrary.GROVE_LCD.REG_ADDR.ENTRYMODESET)
        ' ユーザー定義
        GROVE_LCD.location = 1
        GROVE_LCD.CharMap = New Byte() {&H1F, &H1F, &H1F, &H1F, &H1F, &H1F, &H1F, &H1F}
        Await GROVE_LCD.Write(I2CLibrary.GROVE_LCD.REG_ADDR.SETCGRAMADDR)
        ' SetCursol
        GROVE_LCD.col = 0
        GROVE_LCD.row = 0
        Await GROVE_LCD.Write(I2CLibrary.GROVE_LCD.REG_ADDR.SETDDRAMADDR)
        GROVE_LCD.WriteDelay = 50
        GROVE_LCD.Data = New Byte() {&H1, &H41, &H42, &H43, &H44, &H45, &H46, &H47, &H48, &H49,
        &H4A, &H4B, &H4C, &H4D, &H4E, &H4F}
        Await GROVE_LCD.Write(I2CLibrary.GROVE_LCD.REG_ADDR.PRINT)
    End Sub

    Private Sub btnBack_Click(sender As Object, e As RoutedEventArgs)
        GROVE_LCD.close()
        App.rootFrame.Navigate(GetType(MenuPage))
    End Sub
End Class
```

3-9-2. 液晶バックライト PCA9633

変更履歴

2016年12月05日 第一版 新規作成

1. 概要

液晶バックライト(PCA9633)とI2C 接続するための機能を提供します。

2. 機能一覧

プロパティ

| 機能名 | 機能説明 |
|----------|---------------------------------|
| SLEEP | 低消費電力モード |
| DMBLNK | 0: dimming 1: blinking |
| INVRT | 出力論理状態が反転しない。 |
| OCH | STOP コマンドで出力が変化する |
| OUTDRV | 4 つの LED 出力はトータムポール構造で構成されています。 |
| OUTNE | 出力ドライバの有効 |
| PWM0 | 青 |
| PWM1 | 緑 |
| PWM2 | 赤 |
| PWM3 | -- |
| GRPPWM | DMBLNK=false: 輝度 DMBLNK=1:点滅周期 |
| GRPFREQ | DMBLNK=1:点滅周期 |
| LDR3 | LED3 出力状態制御 |
| LDR2 | LED2 出力状態制御 |
| LDR1 | LED1 出力状態制御 |
| LDR0 | LED0 出力状態制御 |
| ReadBuff | 読込生データ |

メソッド

| 機能名 | 機能説明 |
|-------|---------------------------------------|
| Init | I2C バスコントローラに指定された接続設定を使って非同期的に接続します。 |
| Read | 指定レジスタ読込 |
| Write | 指定レジスタ書込 |

定数

| 機能名 | 機能説明 | 定数值 |
|------------------|--------------------------|------|
| REG_ADDR.MODE1 | Mode register 1 | 0x00 |
| REG_ADDR.MODE2 | Mode register 2 | 0x01 |
| REG_ADDR.PWM0 | brightness control LED0 | 0x02 |
| REG_ADDR.PWM1 | brightness control LED1 | 0x03 |
| REG_ADDR.PWM2 | brightness control LED2 | 0x04 |
| REG_ADDR.PWM3 | brightness control LED3 | 0x05 |
| REG_ADDR.GRPPWM | group duty cycle control | 0x06 |
| REG_ADDR.GRPFREQ | group frequency | 0x07 |
| REG_ADDR.LEDOUT | LED output state | 0x08 |

SLEEP プロパティ

概要

MODE1 レジスタの SLEEP の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.MODE1` を指定した時に保持されます。

Write メソッドで `REG_ADDR.MODE1` を指定した時に SLEEP 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.MODE1` を指定した時に保持した値
または、SET 処理をした値

DMBLNK プロパティ

概要

MODE2 レジスタの DMBLNK の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.MODE2` を指定した時に保持されます。

Write メソッドで `REG_ADDR.MODE2` を指定した時に DMBLNK 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.MODE2` を指定した時に保持した値
または、SET 処理をした値

INVRT プロパティ

概要

MODE2 レジスタの INVRT の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.MODE2` を指定した時に保持されます。

Write メソッドで `REG_ADDR.MODE2` を指定した時に INVRT 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.MODE2` を指定した時に保持した値
または、SET 処理をした値

OCH プロパティ

概要

MODE2 レジスタの OCH の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.MODE2` を指定した時に保持されます。

Write メソッドで `REG_ADDR.MODE2` を指定した時に OCH 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.MODE2` を指定した時に保持した値
または、SET 処理をした値

OUTDRV プロパティ

概要

MODE2 レジスタの OUTDRV の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.MODE2` を指定した時に保持されます。

Write メソッドで `REG_ADDR.MODE2` を指定した時に OUTDRV 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.MODE2` を指定した時に保持した値
または、SET 処理をした値

OUTNE プロパティ

概要

MODE2 レジスタの OUTNE の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.MODE2` を指定した時に保持されます。

Write メソッドで `REG_ADDR.MODE2` を指定した時に OUTNE 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.MODE2` を指定した時に保持した値
または、SET 処理をした値

PWM0 プロパティ

概要

PWM0 レジスタの PWM0 の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.PWM` を指定した時に保持されます。

Write メソッドで `REG_ADDR.PWM` を指定した時に PWM0 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.PWM` を指定した時に保持した値
または、SET 処理をした値

PWM1 プロパティ

概要

PWM1 レジスタの PWM1 の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.PWM` を指定した時に保持されます。

Write メソッドで `REG_ADDR.PWM` を指定した時に PWM1 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.PWM` を指定した時に保持した値
または、SET 処理をした値

PWM2 プロパティ

概要

PWM2 レジスタの PWM2 の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.PWM` を指定した時に保持されます。

Write メソッドで `REG_ADDR.PWM` を指定した時に PWM2 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.PWM` を指定した時に保持した値
または、SET 処理をした値

PWM3 プロパティ

概要

PWM3 レジスタの PWM3 の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.PWM` を指定した時に保持されます。

Write メソッドで `REG_ADDR.PWM` を指定した時に PWM3 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.PWM` を指定した時に保持した値
または、SET 処理をした値

GRPPWM プロパティ

概要

GRPPWM レジスタの GRPPWM の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.GRPPWM` を指定した時に保持されます。

Write メソッドで `REG_ADDR.GRPPWM` を指定した時に GRPPWM 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.GRPPWM` を指定した時に保持した値
または、SET 処理をした値

GRPFREQ プロパティ

概要

GRPFREQ レジスタの GRPFREQ の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.GRPFREQ` を指定した時に保持されます。

Write メソッドで `REG_ADDR.GRPFREQ` を指定した時に GRPFREQ 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.GRPFREQ` を指定した時に保持した値
または、SET 処理をした値

LDR0 プロパティ

概要

LEDOUT レジスタの LDR0 の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.LEDOUT` を指定した時に保持されます。

Write メソッドで `REG_ADDR.LEDOUT` を指定した時に LDR0 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.LEDOUT` を指定した時に保持した値
または、SET 処理をした値

LDR1 プロパティ

概要

LEDOUT レジスタの LDR1 の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.LEDOUT` を指定した時に保持されます。

Write メソッドで `REG_ADDR.LEDOUT` を指定した時に LDR1 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.LEDOUT` を指定した時に保持した値
または、SET 処理をした値

LDR2 プロパティ

概要

LEDOUT レジスタの LDR2 の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.LEDOUT` を指定した時に保持されます。

Write メソッドで `REG_ADDR.LEDOUT` を指定した時に LDR2 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.LEDOUT` を指定した時に保持した値
または、SET 処理をした値

LDR3 プロパティ

概要

LEDOUT レジスタの LDR3 の値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.LEDOUT` を指定した時に保持されます。

Write メソッドで `REG_ADDR.LEDOUT` を指定した時に LDR3 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.LEDOUT` を指定した時に保持した値
または、SET 処理をした値

ReadBuff プロパティ

概要

Read メソッド時に保持されます。

処理内容

Read メソッド時に保持されます。

GET 処理

Read メソッド時に保持した値

Init メソッド

概要

I2C バスコントローラに指定された接続設定を使って非同期的に接続します。

引数

| 引数名 | 型 | 説明 |
|---------|------|-----------|
| Address | byte | I2C のアドレス |

戻り値

なし

Read メソッド

概要

最新の変換結果を読み込みます。

引数

| 引数名 | 型 | 説明 |
|----------|------|-----------|
| REG_ADDR | byte | 0x00～0x07 |

戻り値

Status と同じ

1: 正常

1 以外: 異常

読込処理

| REG_ADDR 値 | 読込プロパティ |
|------------|---|
| MODE1 | SLEEP |
| MODE2 | DMBLNK INVRT OCH OUTDRV OUTNE |
| PWM | PWM0 PWM1 PWM2 PWM3 |
| GRPPWM | GRPPWM |
| GRPFREQ | GRPFREQ |
| LEDOUT | LDR0 LDR1 LDR2 LDR3 |

Write メソッド

概要

引数 REG_ADDR に関するプロパティを編集し書き込みます。

引数

| 引数名 | 型 | 説明 |
|----------|------|-----------|
| REG_ADDR | byte | 0x01～0x07 |

戻り値

Status

1: 正常

1 以外: 異常

書込処理

| REG_ADDR 値 | 書込プロパティ |
|------------|---|
| MODE1 | SLEEP |
| MODE2 | DMBLNK INVRT OCH OUTDRV OUTNE |
| PWM | PWM0 PWM1 PWM2 PWM3 |
| GRPPWM | GRPPWM |
| GRPFREQ | GRPFREQ |
| LEDOUT | LDR0 LDR1 LDR2 LDR3 |

サンプルプログラム

```
Imports I2CLibrary
Public NotInheritable Class PIC16F77APage
    Inherits Page
    Private Const PCA9633_Address As Byte = &H62
    Private PCA9633 As I2CLibrary.PCA9633 = New I2CLibrary.PCA9633()

    Public Sub New()
        InitializeComponent()
        Call Init()
    End Sub
    Private Async Sub Init()
        Await PCA9633.Init(PCA9633_Address)
    End Sub
    Private Sub btnLED_Click(sender As Object, e As RoutedEventArgs)
        PCA9633.SLEEP = False
        PCA9633.Write(I2CLibrary.PCA9633.REG_ADDR.MODE1)

        PCA9633.LDR0 = &H3
        PCA9633.LDR1 = &H3
        PCA9633.LDR2 = &H3
        PCA9633.LDR3 = &H3
        PCA9633.Write(I2CLibrary.PCA9633.REG_ADDR.LEDOUT)

        PCA9633.DMBLNK = True
        PCA9633.INVRT = False
        PCA9633.OCH = False
        PCA9633.OUTDRV = False
        PCA9633.OUTNE = &H0
        PCA9633.Write(I2CLibrary.PCA9633.REG_ADDR.MODE2)

        PCA9633.GRPFREQ = &H10
        PCA9633.Write(I2CLibrary.PCA9633.REG_ADDR.GRPFREQ)

        PCA9633.GRPPWM = &H8
        PCA9633.Write(I2CLibrary.PCA9633.REG_ADDR.GRPPWM)

        PCA9633.PWM0 = &H0      ' 青
        PCA9633.PWM1 = &H0      ' 緑
        PCA9633.PWM2 = &HFF     ' 赤
        PCA9633.PWM3 = &H0
        PCA9633.Write(I2CLibrary.PCA9633.REG_ADDR.PWM)
    End Sub
    Private Sub btnBack_Click(sender As Object, e As RoutedEventArgs)
        PCA9633.close()
        App.rootFrame.Navigate(GetType(MenuPage))
    End Sub
End Class
```

3-10.1 6 c h PWMコントローラ PCA9685

変更履歴

2016年9月23日 第一版 新規作成

1. 概要

16CH、PWMドライバ(PCA9685)とI2C 接続するための機能を提供します。

2. 機能一覧

プロパティ

| 機能名 | 機能説明 |
|------------|---|
| MODE1 | MODE1 レジスタを取得または設定します。 |
| MODE2 | MODE2 レジスタを取得または設定します。 |
| SUBADR1 | I2C-bus subaddress 1 を取得または設定します。 |
| SUBADR2 | I2C-bus subaddress 2 を取得または設定します。 |
| SUBADR3 | I2C-bus subaddress 3 を取得または設定します。 |
| ALLCALLADR | ALLCALL I2C-bus address register を取得または設定します。 |
| LED0_ON | LED0 を ON にするカウントを取得または設定します。 |
| LED0_OFF | LED0 を OFF にするカウントを取得または設定します。 |
| LED1_ON | LED1 を ON にするカウントを取得または設定します。 |
| LED1_OFF | LED1 を OFF にするカウントを取得または設定します。 |
| LED2_ON | LED2 を ON にするカウントを取得または設定します。 |
| LED2_OFF | LED2 を OFF にするカウントを取得または設定します。 |
| LED3_ON | LED3 を ON にするカウントを取得または設定します。 |

| 機能名 | 機能説明 |
|-------------|--------------------------------------|
| LED3_OFF | LED3 を OFF にするカウントを取得または設定します。 |
| LED4_ON | LED4 を ON にするカウントを取得または設定します。 |
| LED4_OFF | LED4 を OFF にするカウントを取得または設定します。 |
| LED5_ON | LED5 を ON にするカウントを取得または設定します。 |
| LED5_OFF | LED5 を OFF にするカウントを取得または設定します。 |
| LED6_ON | LED6 を ON にするカウントを取得または設定します。 |
| LED6_OFF | LED6 を OFF にするカウントを取得または設定します。 |
| LED7_ON | LED7 を ON にするカウントを取得または設定します。 |
| LED7_OFF | LED7 を OFF にするカウントを取得または設定します。 |
| LED8_ON | LED8 を ON にするカウントを取得または設定します。 |
| LED8_OFF | LED8 を OFF にするカウントを取得または設定します。 |
| LED9_ON | LED10 を ON にするカウントを取得または設定します。 |
| LED9_OFF | LED10 を OFF にするカウントを取得または設定します。 |
| LED11_ON | LED11 を ON にするカウントを取得または設定します。 |
| LED11_OFF | LED11 を OFF にするカウントを取得または設定します。 |
| LED12_ON | LED12 を ON にするカウントを取得または設定します。 |
| LED12_OFF | LED12 を OFF にするカウントを取得または設定します。 |
| LED13_ON | LED13 を ON にするカウントを取得または設定します。 |
| LED13_OFF | LED13 を OFF にするカウントを取得または設定します。 |
| LED14_ON | LED14 を ON にするカウントを取得または設定します。 |
| LED14_OFF | LED14 を OFF にするカウントを取得または設定します。 |
| LED15_ON | LED15 を ON にするカウントを取得または設定します。 |
| LED15_OFF | LED15 を OFF にするカウントを取得または設定します。 |
| ALL_LED_ON | LED0~LED15 を ON にするカウントを取得または設定します。 |
| ALL_LED_OFF | LED0~LED15 を OFF にするカウントを取得または設定します。 |
| PRE_SCALE | PRE_SCALE レジスタを取得または設定します。 |
| TestMode | TestMode レジスタを取得または設定します。 |

メソッド

| 機能名 | 機能説明 |
|-------|---------------------------------------|
| Init | I2C バスコントローラに指定された接続設定を使って非同期的に接続します。 |
| Read | 指定レジスタ読込 |
| Write | 指定レジスタ書込 |

定数

| 機能名 | 機能説明 | 定数值 |
|---------------------|-------------------|------|
| REG_ADDR.MODE1 | レジスタ MODE1 | 0x00 |
| REG_ADDR.MODE2 | レジスタ MODE2 | 0x01 |
| REG_ADDR.SUBADR1 | レジスタ SUBADR1 | 0x18 |
| REG_ADDR.ALLCALLADR | レジスタ ALLCALLADR | 0x1B |
| REG_ADDR.LED0 | レジスタ LED0_ON_L | 0x06 |
| REG_ADDR.LED1 | レジスタ LED1_ON_L | 0x0A |
| REG_ADDR.LED2 | レジスタ LED2_ON_L | 0x0E |
| REG_ADDR.LED3 | レジスタ LED3_ON_L | 0x12 |
| REG_ADDR.LED4 | レジスタ LED4_ON_L | 0x16 |
| REG_ADDR.LED5 | レジスタ LED5_ON_L | 0x1A |
| REG_ADDR.LED6 | レジスタ LED6_ON_L | 0x1E |
| REG_ADDR.LED7 | レジスタ LED7_ON_L | 0x22 |
| REG_ADDR.LED8 | レジスタ LED8_ON_L | 0x26 |
| REG_ADDR.LED9 | レジスタ LED9_ON_L | 0x2A |
| REG_ADDR.LED10 | レジスタ LED10_ON_L | 0x2E |
| REG_ADDR.LED11 | レジスタ LED11_ON_L | 0x32 |
| REG_ADDR.LED12 | レジスタ LED12_ON_L | 0x36 |
| REG_ADDR.LED13 | レジスタ LED13_ON_L | 0x3A |
| REG_ADDR.LED14 | レジスタ LED14_ON_L | 0x3E |
| REG_ADDR.LED15 | レジスタ LED15_ON_L | 0x42 |
| REG_ADDR.ALL_LED | レジスタ ALL_LED_ON_L | 0xFA |
| REG_ADDR.PRE_SCALE | レジスタ PRE_SCALE | 0xFE |
| REG_ADDR.TestMode | レジスタ TestMode | 0xFF |

MODE1 プロパティ

概要

MODE1 レジスタを取得または設定します。

処理内容

Read メソッドで `REG_ADDR.MODE1` を指定した時に MODE1 レジスタが保持されます。

Write メソッドで `REG_ADDR.MODE1` を指定した時に MODE1 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.MODE1` を指定した時に保持した値
または、SET 処理をした値

MODE2 プロパティ

概要

MODE2 レジスタを取得または設定します。

処理内容

Read メソッドで `REG_ADDR.MODE2` を指定した時に MODE2 レジスタが保持されます。

Write メソッドで `REG_ADDR.MODE2` を指定した時に MODE2 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.MODE2` を指定した時に保持した値
または、SET 処理をした値

SUBADR1 プロパティ

概要

I2C-bus subaddress 1 を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.SUBADR1` を指定した時に SUBADR1 レジスタの値が保持されます。

Write メソッドで `REG_ADDR.SUBADR1` を指定した時に SUBADR1 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.SUBADR1` を指定した時に保持した値
または、SET 処理をした値

SUBADR2 プロパティ

概要

I2C-bus subaddress 2 を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.SUBADR1` を指定した時に SUBADR2 レジスタの値が保持されます。

Write メソッドで `REG_ADDR.SUBADR1` を指定した時に SUBADR2 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.SUBADR1` を指定した時に保持した値
または、SET 処理をした値

SUBADR3 プロパティ

概要

I2C-bus subaddress 3 を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.SUBADR1` を指定した時に SUBADR3 レジスタの値が保持されます。

Write メソッドで `REG_ADDR.SUBADR1` を指定した時に SUBADR3 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.SUBADR1` を指定した時に保持した値
または、SET 処理をした値

ALLCALLADR プロパティ

概要

ALLCALL I2C-bus address register を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.ALLCALLADR` を指定した時に ALLCALLADR レジスタの値が保持されます。

Write メソッドで `REG_ADDR.ALLCALLADR` を指定した時に ALLCALLADR 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.ALLCALLADR` を指定した時に保持した値
または、SET 処理をした値

LED0_ON プロパティ

概要

LED0 を ON にするカウントを取得または設定します。

処理内容

Read メソッドで `REG_ADDR.LED0` を指定した時に `LED0_ON_L`、`LED0_ON_H` レジスタの値が `LED0_ON` に保持されます。

Write メソッドで `REG_ADDR.LED0` を指定した時に `LED0_ON` 値を使用して書込みます。

有効値 (0~13 ビット)

1~12 ビット: 0~4095 (カウント)

13 ビット: 1 の場合 (4096 以上の場合) (Full On)

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.LED0` を指定した時に保持した値
または、SET 処理をした値

LED0_OFF プロパティ

概要

LED0 を OFF にするカウントを取得または設定します。

処理内容

Read メソッドで `REG_ADDR.LED0` を指定した時に `LED0_OFF_L`、`LED0_OFF_H` レジスタの値が `LED0_OFF` に保持されます。

Write メソッドで `REG_ADDR.LED0` を指定した時に `LED0_OFF` 値を使用して書込みます。

有効値 (0~13 ビット)

1~12 ビット: 0~4095 (カウント)

13 ビット: 1 の場合 (4096 以上の場合) (Full Off)

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.LED0` を指定した時に保持した値
または、SET 処理をした値

LED1_ON プロパティ

概要

LED1 を ON にするカウントを取得または設定します。

処理内容

Read メソッドで `REG_ADDR.LED1` を指定した時に `LED1_ON_L`、`LED1_ON_H` レジスタの値が `LED1_ON` に保持されます。

Write メソッドで `REG_ADDR.LED1` を指定した時に `LED1_ON` 値を使用して書込みます。

有効値 (0~13 ビット)

1~12 ビット: 0~4095 (カウント)

13 ビット: 1 の場合 (4096 以上の場合) (Full On)

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.LED1` を指定した時に保持した値
または、SET 処理をした値

LED1_OFF プロパティ

概要

LED1 を OFF にするカウントを取得または設定します。

処理内容

Read メソッドで `REG_ADDR.LED1` を指定した時に `LED1_OFF_L`、`LED1_OFF_H` レジスタの値が `LED1_OFF` に保持されます。

Write メソッドで `REG_ADDR.LED1` を指定した時に `LED1_OFF` 値を使用して書込みます。

有効値 (0~13 ビット)

1~12 ビット: 0~4095 (カウント)

13 ビット: 1 の場合 (4096 以上の場合) (Full Off)

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.LED1` を指定した時に保持した値
または、SET 処理をした値

LED2_ON プロパティ

概要

LED2 を ON にするカウントを取得または設定します。

処理内容

Read メソッドで `REG_ADDR.LED2` を指定した時に `LED2_ON_L`、`LED2_ON_H` レジスタの値が `LED2_ON` に保持されます。

Write メソッドで `REG_ADDR.LED2` を指定した時に `LED2_ON` 値を使用して書込みます。

有効値 (0~13 ビット)

1~12 ビット: 0~4095 (カウント)

13 ビット: 1 の場合 (4096 以上の場合) (Full On)

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.LED2` を指定した時に保持した値
または、SET 処理をした値

LED2_OFF プロパティ

概要

LED2 を OFF にするカウントを取得または設定します。

処理内容

Read メソッドで `REG_ADDR.LED2` を指定した時に `LED2_OFF_L`、`LED2_OFF_H` レジスタの値が `LED2_OFF` に保持されます。

Write メソッドで `REG_ADDR.LED2` を指定した時に `LED2_OFF` 値を使用して書込みます。

有効値 (0~13 ビット)

1~12 ビット: 0~4095 (カウント)

13 ビット: 1 の場合 (4096 以上の場合) (Full Off)

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.LED2` を指定した時に保持した値
または、SET 処理をした値

LED3_ON プロパティ

概要

LED3 を ON にするカウントを取得または設定します。

処理内容

Read メソッドで `REG_ADDR.LED3` を指定した時に `LED3_ON_L`、`LED3_ON_H` レジスタの値が `LED3_ON` に保持されます。

Write メソッドで `REG_ADDR.LED3` を指定した時に `LED2_ON` 値を使用して書込みます。

有効値 (0~13 ビット)

1~12 ビット: 0~4095 (カウント)

13 ビット: 1 の場合 (4096 以上の場合) (Full On)

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.LED3` を指定した時に保持した値
または、SET 処理をした値

LED3_OFF プロパティ

概要

LED3 を OFF にするカウントを取得または設定します。

処理内容

Read メソッドで `REG_ADDR.LED3` を指定した時に `LED3_OFF_L`、`LED3_OFF_H` レジスタの値が `LED3_OFF` に保持されます。

Write メソッドで `REG_ADDR.LED3` を指定した時に `LED3_OFF` 値を使用して書込みます。

有効値 (0~13 ビット)

1~12 ビット: 0~4095 (カウント)

13 ビット: 1 の場合 (4096 以上の場合) (Full Off)

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.LED3` を指定した時に保持した値
または、SET 処理をした値

LED4_ON プロパティ

概要

LED4 を ON にするカウントを取得または設定します。

処理内容

Read メソッドで `REG_ADDR.LED4` を指定した時に `LED4_ON_L`、`LED4_ON_H` レジスタの値が `LED4_ON` に保持されます。

Write メソッドで `REG_ADDR.LED4` を指定した時に `LED4_ON` 値を使用して書込みます。

有効値 (0~13 ビット)

1~12 ビット: 0~4095 (カウント)

13 ビット: 1 の場合 (4096 以上の場合) (Full On)

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.LED4` を指定した時に保持した値
または、SET 処理をした値

LED4_OFF プロパティ

概要

LED4 を OFF にするカウントを取得または設定します。

処理内容

Read メソッドで `REG_ADDR.LED4` を指定した時に `LED4_OFF_L`、`LED4_OFF_H` レジスタの値が `LED4_OFF` に保持されます。

Write メソッドで `REG_ADDR.LED4` を指定した時に `LED4_OFF` 値を使用して書込みます。

有効値 (0~13 ビット)

1~12 ビット: 0~4095 (カウント)

13 ビット: 1 の場合 (4096 以上の場合) (Full Off)

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.LED4` を指定した時に保持した値
または、SET 処理をした値

LED5_ON プロパティ

概要

LED5 を ON にするカウントを取得または設定します。

処理内容

Read メソッドで `REG_ADDR.LED5` を指定した時に `LED5_ON_L`、`LED5_ON_H` レジスタの値が `LED5_ON` に保持されます。

Write メソッドで `REG_ADDR.LED5` を指定した時に `LED5_ON` 値を使用して書込みます。

有効値 (0~13 ビット)

1~12 ビット: 0~4095 (カウント)

13 ビット: 1 の場合 (4096 以上の場合) (Full On)

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.LED5` を指定した時に保持した値
または、SET 処理をした値

LED5_OFF プロパティ

概要

LED5 を OFF にするカウントを取得または設定します。

処理内容

Read メソッドで `REG_ADDR.LED5` を指定した時に `LED5_OFF_L`、`LED5_OFF_H` レジスタの値が `LED5_OFF` に保持されます。

Write メソッドで `REG_ADDR.LED5` を指定した時に `LED5_OFF` 値を使用して書込みます。

有効値 (0~13 ビット)

1~12 ビット: 0~4095 (カウント)

13 ビット: 1 の場合 (4096 以上の場合) (Full Off)

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.LED5` を指定した時に保持した値
または、SET 処理をした値

LED6_ON プロパティ

概要

LED6 を ON にするカウントを取得または設定します。

処理内容

Read メソッドで `REG_ADDR.LED6` を指定した時に `LED6_ON_L`、`LED6_ON_H` レジスタの値が `LED6_ON` に保持されます。

Write メソッドで `REG_ADDR.LED6` を指定した時に `LED6_ON` 値を使用して書込みます。

有効値 (0~13 ビット)

1~12 ビット: 0~4095 (カウント)

13 ビット: 1 の場合 (4096 以上の場合) (Full On)

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.LED6` を指定した時に保持した値
または、SET 処理をした値

LED6_OFF プロパティ

概要

LED6 を OFF にするカウントを取得または設定します。

処理内容

Read メソッドで `REG_ADDR.LED6` を指定した時に `LED6_OFF_L`、`LED6_OFF_H` レジスタの値が `LED6_OFF` に保持されます。

Write メソッドで `REG_ADDR.LED6` を指定した時に `LED6_OFF` 値を使用して書込みます。

有効値 (0~13 ビット)

1~12 ビット: 0~4095 (カウント)

13 ビット: 1 の場合 (4096 以上の場合) (Full Off)

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.LED6` を指定した時に保持した値
または、SET 処理をした値

LED7_ON プロパティ

概要

LED7 を ON にするカウントを取得または設定します。

処理内容

Read メソッドで `REG_ADDR.LED7` を指定した時に `LED7_ON_L`、`LED7_ON_H` レジスタの値が `LED7_ON` に保持されます。

Write メソッドで `REG_ADDR.LED7` を指定した時に `LED7_ON` 値を使用して書込みます。

有効値 (0~13 ビット)

1~12 ビット: 0~4095 (カウント)

13 ビット: 1 の場合 (4096 以上の場合) (Full On)

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.LED7` を指定した時に保持した値
または、SET 処理をした値

LED7_OFF プロパティ

概要

LED7 を OFF にするカウントを取得または設定します。

処理内容

Read メソッドで `REG_ADDR.LED7` を指定した時に `LED7_OFF_L`、`LED7_OFF_H` レジスタの値が `LED7_OFF` に保持されます。

Write メソッドで `REG_ADDR.LED7` を指定した時に `LED7_OFF` 値を使用して書込みます。

有効値 (0~13 ビット)

1~12 ビット: 0~4095 (カウント)

13 ビット: 1 の場合 (4096 以上の場合) (Full Off)

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.LED7` を指定した時に保持した値
または、SET 処理をした値

LED8_ON プロパティ

概要

LED8 を ON にするカウントを取得または設定します。

処理内容

Read メソッドで `REG_ADDR.LED8` を指定した時に `LED8_ON_L`、`LED8_ON_H` レジスタの値が `LED8_ON` に保持されます。

Write メソッドで `REG_ADDR.LED8` を指定した時に `LED8_ON` 値を使用して書込みます。

有効値 (0~13 ビット)

1~12 ビット: 0~4095 (カウント)

13 ビット: 1 の場合 (4096 以上の場合) (Full On)

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.LED8` を指定した時に保持した値
または、SET 処理をした値

LED8_OFF プロパティ

概要

LED8 を OFF にするカウントを取得または設定します。

処理内容

Read メソッドで `REG_ADDR.LED8` を指定した時に `LED8_OFF_L`、`LED8_OFF_H` レジスタの値が `LED8_OFF` に保持されます。

Write メソッドで `REG_ADDR.LED8` を指定した時に `LED8_OFF` 値を使用して書込みます。

有効値 (0~13 ビット)

1~12 ビット: 0~4095 (カウント)

13 ビット: 1 の場合 (4096 以上の場合) (Full Off)

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.LED8` を指定した時に保持した値
または、SET 処理をした値

LED9_ON プロパティ

概要

LED9 を ON にするカウントを取得または設定します。

処理内容

Read メソッドで `REG_ADDR.LED9` を指定した時に `LED9_ON_L`、`LED9_ON_H` レジスタの値が `LED9_ON` に保持されます。

Write メソッドで `REG_ADDR.LED9` を指定した時に `LED9_ON` 値を使用して書込みます。

有効値 (0~13 ビット)

1~12 ビット: 0~4095 (カウント)

13 ビット: 1 の場合 (4096 以上の場合) (Full On)

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.LED9` を指定した時に保持した値
または、SET 処理をした値

LED9_OFF プロパティ

概要

LED9 を OFF にするカウントを取得または設定します。

処理内容

Read メソッドで `REG_ADDR.LED9` を指定した時に `LED9_OFF_L`、`LED9_OFF_H` レジスタの値が `LED9_OFF` に保持されます。

Write メソッドで `REG_ADDR.LED9` を指定した時に `LED9_OFF` 値を使用して書込みます。

有効値 (0~13 ビット)

1~12 ビット: 0~4095 (カウント)

13 ビット: 1 の場合 (4096 以上の場合) (Full Off)

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.LED9` を指定した時に保持した値
または、SET 処理をした値

LED10_ON プロパティ

概要

LED10 を ON にするカウントを取得または設定します。

処理内容

Read メソッドで `REG_ADDR.LED10` を指定した時に `LED10_ON_L`、`LED10_ON_H` レジスタの値が `LED10_ON` に保持されます。

Write メソッドで `REG_ADDR.LED10` を指定した時に `LED10_ON` 値を使用して書込みます。

有効値 (0~13 ビット)

1~12 ビット: 0~4095 (カウント)

13 ビット: 1 の場合 (4096 以上の場合) (Full On)

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.LED10` を指定した時に保持した値
または、SET 処理をした値

LED10_OFF プロパティ

概要

LED10 を OFF にするカウントを取得または設定します。

処理内容

Read メソッドで `REG_ADDR.LED10` を指定した時に `LED10_OFF_L`、`LED10_OFF_H` レジスタの値が `LED10_OFF` に保持されます。

Write メソッドで `REG_ADDR.LED10` を指定した時に `LED10_OFF` 値を使用して書込ます。

有効値 (0~13 ビット)

1~12 ビット: 0~4095 (カウント)

13 ビット: 1 の場合 (4096 以上の場合) (Full Off)

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.LED10` を指定した時に保持した値
または、SET 処理をした値

LED11_ON プロパティ

概要

LED11 を ON にするカウントを取得または設定します。

処理内容

Read メソッドで `REG_ADDR.LED11` を指定した時に `LED11_ON_L`、`LED11_ON_H` レジスタの値が `LED11_ON` に保持されます。

Write メソッドで `REG_ADDR.LED11` を指定した時に `LED11_ON` 値を使用して書込みます。

有効値 (0~13 ビット)

1~12 ビット: 0~4095 (カウント)

13 ビット: 1 の場合 (4096 以上の場合) (Full On)

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.LED11` を指定した時に保持した値
または、SET 処理をした値

LED11_OFF プロパティ

概要

LED11 を OFF にするカウントを取得または設定します。

処理内容

Read メソッドで `REG_ADDR.LED11` を指定した時に `LED11_OFF_L`、`LED11_OFF_H` レジスタの値が `LED11_OFF` に保持されます。

Write メソッドで `REG_ADDR.LED11` を指定した時に `LED11_OFF` 値を使用して書込ます。

有効値 (0~13 ビット)

1~12 ビット: 0~4095 (カウント)

13 ビット: 1 の場合 (4096 以上の場合) (Full Off)

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.LED11` を指定した時に保持した値
または、SET 処理をした値

LED12_ON プロパティ

概要

LED12 を ON にするカウントを取得または設定します。

処理内容

Read メソッドで `REG_ADDR.LED12` を指定した時に `LED12_ON_L`、`LED12_ON_H` レジスタの値が `LED12_ON` に保持されます。

Write メソッドで `REG_ADDR.LED12` を指定した時に `LED12_ON` 値を使用して書込みます。

有効値 (0~13 ビット)

1~12 ビット: 0~4095 (カウント)

13 ビット: 1 の場合 (4096 以上の場合) (Full On)

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.LED12` を指定した時に保持した値
または、SET 処理をした値

LED12_OFF プロパティ

概要

LED12 を OFF にするカウントを取得または設定します。

処理内容

Read メソッドで `REG_ADDR.LED12` を指定した時に `LED12_OFF_L`、`LED12_OFF_H` レジスタの値が `LED12_OFF` に保持されます。

Write メソッドで `REG_ADDR.LED12` を指定した時に `LED12_OFF` 値を使用して書込ます。

有効値 (0~13 ビット)

1~12 ビット: 0~4095 (カウント)

13 ビット: 1 の場合 (4096 以上の場合) (Full Off)

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.LED12` を指定した時に保持した値
または、SET 処理をした値

LED13_ON プロパティ

概要

LED13 を ON にするカウントを取得または設定します。

処理内容

Read メソッドで `REG_ADDR.LED13` を指定した時に `LED13_ON_L`、`LED13_ON_H` レジスタの値が `LED13_ON` に保持されます。

Write メソッドで `REG_ADDR.LED13` を指定した時に `LED13_ON` 値を使用して書込みます。

有効値 (0~13 ビット)

1~12 ビット: 0~4095 (カウント)

13 ビット: 1 の場合 (4096 以上の場合) (Full On)

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.LED13` を指定した時に保持した値
または、SET 処理をした値

LED13_OFF プロパティ

概要

LED13 を OFF にするカウントを取得または設定します。

処理内容

Read メソッドで `REG_ADDR.LED13` を指定した時に `LED13_OFF_L`、`LED13_OFF_H` レジスタの値が `LED13_OFF` に保持されます。

Write メソッドで `REG_ADDR.LED13` を指定した時に `LED13_OFF` 値を使用して書込ます。

有効値 (0~13 ビット)

1~12 ビット: 0~4095 (カウント)

13 ビット: 1 の場合 (4096 以上の場合) (Full Off)

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.LED13` を指定した時に保持した値
または、SET 処理をした値

LED14_ON プロパティ

概要

LED14 を ON にするカウントを取得または設定します。

処理内容

Read メソッドで `REG_ADDR.LED14` を指定した時に `LED14_ON_L`、`LED14_ON_H` レジスタの値が `LED14_ON` に保持されます。

Write メソッドで `REG_ADDR.LED14` を指定した時に `LED14_ON` 値を使用して書込みます。

有効値 (0~13 ビット)

1~12 ビット: 0~4095 (カウント)

13 ビット: 1 の場合 (4096 以上の場合) (Full On)

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.LED14` を指定した時に保持した値
または、SET 処理をした値

LED14_OFF プロパティ

概要

LED14 を OFF にするカウントを取得または設定します。

処理内容

Read メソッドで `REG_ADDR.LED14` を指定した時に `LED14_OFF_L`、`LED14_OFF_H` レジスタの値が `LED14_OFF` に保持されます。

Write メソッドで `REG_ADDR.LED14` を指定した時に `LED14_OFF` 値を使用して書込ます。

有効値 (0~13 ビット)

1~12 ビット: 0~4095 (カウント)

13 ビット: 1 の場合 (4096 以上の場合) (Full Off)

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.LED14` を指定した時に保持した値
または、SET 処理をした値

LED15_ON プロパティ

概要

LED15 を ON にするカウントを取得または設定します。

処理内容

Read メソッドで `REG_ADDR.LED15` を指定した時に `LED15_ON_L`、`LED15_ON_H` レジスタの値が `LED15_ON` に保持されます。

Write メソッドで `REG_ADDR.LED15` を指定した時に `LED15_ON` 値を使用して書込みます。

有効値 (0~13 ビット)

1~12 ビット: 0~4095 (カウント)

13 ビット: 1 の場合 (4096 以上の場合) (Full On)

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.LED15` を指定した時に保持した値
または、SET 処理をした値

LED15_OFF プロパティ

概要

LED15 を OFF にするカウントを取得または設定します。

処理内容

Read メソッドで `REG_ADDR.LED15` を指定した時に `LED15_OFF_L`、`LED15_OFF_H` レジスタの値が `LED15_OFF` に保持されます。

Write メソッドで `REG_ADDR.LED15` を指定した時に `LED15_OFF` 値を使用して書込ます。

有効値 (0~13 ビット)

1~12 ビット: 0~4095 (カウント)

13 ビット: 1 の場合 (4096 以上の場合) (Full Off)

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.LED15` を指定した時に保持した値
または、SET 処理をした値

ALL_LED_ON プロパティ

概要

LED0～LED15 を ON にするカウントを取得または設定します。

処理内容

Read メソッドで `REG_ADDR.ALL_LED` を指定した時に `ALL_LED_ON_L`、`ALL_LED_ON_H` レジスタの値が `ALL_LED_ON` に保持されます。

Write メソッドで `REG_ADDR.ALL_LED` を指定した時に `ALL_LED_ON` 値を使用して書込みます。

有効値 (0～13 ビット)

1～12 ビット: 0～4095 (カウント)

13 ビット: 1 の場合 (4096 以上の場合) (Full On)

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.ALL_LED` を指定した時に保持した値
または、SET 処理をした値

ALL_LED_OFF プロパティ

概要

LED0～LED15 を OFF にするカウントを取得または設定します。

処理内容

Read メソッドで `REG_ADDR.ALL_LED` を指定した時に `ALL_LED_OFF_L`、`ALL_LED_OFF_H` レジスタの値が `ALL_LED_OFF` に保持されます。

Write メソッドで `REG_ADDR.ALL_LED` を指定した時に `ALL_LED_OFF` 値を使用して書込まず。

有効値 (0～13 ビット)

1～12 ビット: 0～4095 (カウント)

13 ビット: 1 の場合 (4096 以上の場合) (Full Off)

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.ALL_LED` を指定した時に保持した値
または、SET 処理をした値

PRE_SCALE プロパティ

概要

PRE_SCALE レジスタを取得または設定します。

処理内容

Read メソッドで `REG_ADDR.PRE_SCALE` を指定した時に PRE_SCALE レジスタの値が PRE_SCALE に保持されます。

Write メソッドで `REG_ADDR.PRE_SCALE` を指定した時に PRE_SCALE 値を使用して書込みます。

初期値: 0x1E (200Hz)

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.PRE_SCALE` を指定した時に保持した値
または、SET 処理をした値

TestMode プロパティ

概要

TestMode レジスタを取得または設定します。

処理内容

Read メソッドで REG_ [REG_ADDR.TestMode](#) を指定した時に TestMode レジスタの値が TestMode に保持されます。

Write メソッドで REG_ [REG_ADDR.TestMode](#) を指定した時に TestMode 値を使用して書込ます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで REG_ [REG_ADDR.TestMode](#) を指定した時に保持した値
または、SET 処理をした値

Init メソッド

概要

I2C バスコントローラに指定された接続設定を使って非同期的に接続します。

引数

| 引数名 | 型 | 説明 |
|---------|------|-----------|
| Address | byte | I2C のアドレス |

戻り値

なし

Read メソッド

概要

最新の変換結果を読込みます。

引数

| 引数名 | 型 | 説明 |
|----------|------|--|
| REG_ADDR | byte | REG_ADDR_MODE1 (0x00) ~ REG_ADDR_TestMode (0xFF) |

戻り値

Status と同じ

1: 正常

1 以外: 異常

読込処理

| REG_ADDR 値 | 読込プロパティ |
|------------|-------------------------------|
| MODE1 | MODE1 |
| MODE2 | MODE2 |
| SUBADR1 | SUBADR1 SUBADR2 SUBADR3 |
| ALLCALLADR | ALLCALLADR |
| LED0 | LED0_ON LED0_OFF |
| LED1 | LED1_ON LED1_OFF |
| LED2 | LED2_ON LED2_OFF |
| LED3 | LED3_ON LED3_OFF |
| LED4 | LED4_ON LED4_OFF |
| LED5 | LED5_ON LED5_OFF |
| LED6 | LED6_ON LED6_OFF |
| LED7 | LED7_ON LED7_OFF |

| REG_ADDR 値 | 読込プロパティ |
|------------|---------------------------|
| LED8 | LED8_ON LED8_OFF |
| LED9 | LED9_ON LED9_OFF |
| LED10 | LED10_ON LED10_OFF |
| LED11 | LED11_ON LED11_OFF |
| LED12 | LED12_ON LED12_OFF |
| LED13 | LED13_ON LED13_OFF |
| LED14 | LED14_ON LED14_OFF |
| LED15 | LED15_ON LED15_OFF |
| ALL_LED | ALL_LED_ON ALL_LED_OFF |
| PRE_SCALE | PRE_SCALE |
| TestMode | TestMode |

Write メソッド

概要

引数 REG_ADDR に関するプロパティを編集し書き込みます。

引数

| 引数名 | 型 | 説明 |
|----------|------|--|
| REG_ADDR | byte | REG_ADDR_MODE1 (0x00) ~ REG_ADDR_TestMode (0xFF) |

戻り値

Status

1: 正常

1 以外: 異常

書込処理

| REG_ADDR 値 | 書込プロパティ |
|------------|-------------------------------|
| MODE1 | MODE1 |
| MODE2 | MODE2 |
| SUBADR1 | SUBADR1 SUBADR2 SUBADR3 |
| ALLCALLADR | ALLCALLADR |
| LED0 | LED0_ON LED0_OFF |
| LED1 | LED1_ON LED1_OFF |
| LED2 | LED2_ON LED2_OFF |
| LED3 | LED3_ON LED3_OFF |
| LED4 | LED4_ON LED4_OFF |
| LED5 | LED5_ON LED5_OFF |
| LED6 | LED6_ON LED6_OFF |
| LED7 | LED7_ON LED7_OFF |

| REG_ADDR 値 | 書込プロパティ |
|------------|---------------------------|
| LED8 | LED8_ON LED8_OFF |
| LED9 | LED9_ON LED9_OFF |
| LED10 | LED10_ON LED10_OFF |
| LED11 | LED11_ON LED11_OFF |
| LED12 | LED12_ON LED12_OFF |
| LED13 | LED13_ON LED13_OFF |
| LED14 | LED14_ON LED14_OFF |
| LED15 | LED15_ON LED15_OFF |
| ALL_LED | ALL_LED_ON ALL_LED_OFF |
| PRE_SCALE | PRE_SCALE |
| TestMode | TestMode |

サンプルプログラム

```
Imports I2CLibrary
Public NotInheritable Class PCA9685Page
    Inherits Page

    Private Const PCA9685_Address As Byte = &H40
    Private PCA9685 As PCA9685 = New PCA9685()

    Private dOscillator As Double = 25000000      ' オシレータクロック 25MHz
    Private dResolution As Double = 4096         ' 分解能
    Private dCycleTime As Double = 20           ' SG90 サイクルタイム
    Private dHz As Double = 50                  ' SG90 周波数

    Private dMinDutyCycle As Double = 0.5       ' ms
    Private dMaxDutyCycle As Double = 2.4       ' ms
    Private dStd As Double = dResolution / dCycleTime

    Public Sub New()
        InitializeComponent()
        Call Init()
    End Sub

    Private Async Sub Init()
        Await PCA9685.Init(PCA9685_Address)

        Dim iDutyCycle As Integer = 0

        PCA9685.MODE1 = &H31
        PCA9685.Write(I2CLibrary.PCA9685.REG_ADDR.MODE1)

        PCA9685.PRE_SCALE = (dOscillator / (dResolution * dHz)) - 1
        PCA9685.Write(I2CLibrary.PCA9685.REG_ADDR.PRE_SCALE)

        PCA9685.MODE1 = &H21
        PCA9685.Write(I2CLibrary.PCA9685.REG_ADDR.MODE1)

    End Sub

    Private Sub btnHome_Click(sender As Object, e As RoutedEventArgs)
        Dim iDutyCycle As Integer = 0

        iDutyCycle = ((dMaxDutyCycle - dMinDutyCycle) / 2 + dMinDutyCycle) * dStd

        PCA9685.LED0_ON = 0
        PCA9685.LED0_OFF = iDutyCycle

        PCA9685.Write(I2CLibrary.PCA9685.REG_ADDR.LED0)
    End Sub

    Private Sub btnLeft_Click(sender As Object, e As RoutedEventArgs)
        Dim iDutyCycle As Integer = 0
    End Sub
End Class
```

```
iDutyCycle = dMinDutyCycle * dStd

PCA9685.LED0_ON = 0
PCA9685.LED0_OFF = iDutyCycle

PCA9685.Write(I2CLibrary.PCA9685.REG_ADDR.LED0)
End Sub

Private Sub btnRight_Click(sender As Object, e As RoutedEventArgs)
    Dim iDutyCycle As Integer = 0

    iDutyCycle = dMaxDutyCycle * dStd

    PCA9685.LED0_ON = 0
    PCA9685.LED0_OFF = iDutyCycle

    PCA9685.Write(I2CLibrary.PCA9685.REG_ADDR.LED0)
End Sub

Private Sub btnBack_Click(sender As Object, e As RoutedEventArgs)
    PCA9685.close()
    App.rootFrame.Navigate(GetType(MenuPage))
End Sub

End Class
```

3-11.8×8マトリックス LED コントローラ PCA9622

変更履歴

2016年11月24日 第一版 新規作成

1. 概要

8x8 LED マトリックス基板 (PCA9622) と I2C 接続するための機能を提供します。

2. 機能一覧

プロパティ

| 機能名 | 機能説明 |
|-------|--------------------------|
| MODE1 | MODE1 レジスタを取得または設定します。 |
| MODE2 | MODE2 レジスタを取得または設定します。 |
| PWM0 | LED0 の輝度制御値を取得または設定します。 |
| PWM1 | LED1 の輝度制御値を取得または設定します。 |
| PWM2 | LED2 の輝度制御値を取得または設定します。 |
| PWM3 | LED3 の輝度制御値を取得または設定します。 |
| PWM4 | LED4 の輝度制御値を取得または設定します。 |
| PWM5 | LED5 の輝度制御値を取得または設定します。 |
| PWM6 | LED6 の輝度制御値を取得または設定します。 |
| PWM7 | LED7 の輝度制御値を取得または設定します。 |
| PWM8 | LED8 の輝度制御値を取得または設定します。 |
| PWM9 | LED9 の輝度制御値を取得または設定します。 |
| PWM10 | LED10 の輝度制御値を取得または設定します。 |

| 機能名 | 機能説明 |
|------------|---|
| PWM11 | LED11 の輝度制御値を取得または設定します。 |
| PWM12 | LED12 の輝度制御値を取得または設定します。 |
| PWM13 | LED13 の輝度制御値を取得または設定します。 |
| PWM14 | LED14 の輝度制御値を取得または設定します。 |
| PWM15 | LED15 の輝度制御値を取得または設定します。 |
| GRPPWM | グループ輝度制御レジスタを取得または設定します。 |
| GRPFREQ | グループ周波数レジスタ(点滅周期)を取得または設定します。 |
| LDR0 | LED0 の出力制御を取得または設定します。 |
| LDR1 | LED1 の出力制御を取得または設定します。 |
| LDR2 | LED2 の出力制御を取得または設定します。 |
| LDR3 | LED3 の出力制御を取得または設定します。 |
| LDR4 | LED4 の出力制御を取得または設定します。 |
| LDR5 | LED5 の出力制御を取得または設定します。 |
| LDR6 | LED6 の出力制御を取得または設定します。 |
| LDR7 | LED7 の出力制御を取得または設定します。 |
| LDR8 | LED8 の出力制御を取得または設定します。 |
| LDR9 | LED9 の出力制御を取得または設定します。 |
| LDR10 | LED10 の出力制御を取得または設定します。 |
| LDR11 | LED11 の出力制御を取得または設定します。 |
| LDR12 | LED12 の出力制御を取得または設定します。 |
| LDR13 | LED13 の出力制御を取得または設定します。 |
| LDR14 | LED14 の出力制御を取得または設定します。 |
| LDR15 | LED15 の出力制御を取得または設定します。 |
| SUBADR1 | I2C-bus subaddress 1 を取得または設定します。 |
| SUBADR2 | I2C-bus subaddress 2 を取得または設定します。 |
| SUBADR3 | I2C-bus subaddress 3 を取得または設定します。 |
| ALLCALLADR | ALLCALL I2C-bus address register を取得または設定します。 |

メソッド

| 機能名 | 機能説明 |
|-------|---------------------------------------|
| Init | I2C バスコントローラに指定された接続設定を使って非同期的に接続します。 |
| Read | 指定レジスタ読込 |
| Write | 指定レジスタ書込 |

定数

| 機能名 | 機能説明 | 定数值 |
|---------------------|-----------------|------|
| REG_ADDR.MODE1 | レジスタ MODE1 | 0x00 |
| REG_ADDR.MODE2 | レジスタ MODE2 | 0x01 |
| REG_ADDR.PWM0 | レジスタ PWM0 | 0x02 |
| REG_ADDR.GRPPWM | レジスタ GRPPWM | 0x12 |
| REG_ADDR.LEDOUT0 | レジスタ LEDOUT0 | 0x14 |
| REG_ADDR.SUBADR1 | レジスタ SUBADR1 | 0x18 |
| REG_ADDR.ALLCALLADR | レジスタ ALLCALLADR | 0x1B |

MODE1 プロパティ

概要

MODE1 レジスタを取得または設定します。

処理内容

Read メソッドで `REG_ADDR.MODE1` を指定した時に MODE1 レジスタが保持されます。

Write メソッドで `REG_ADDR.MODE1` を指定した時に MODE1 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.MODE1` を指定した時に保持した値
または、SET 処理をした値

MODE2 プロパティ

概要

MODE2 レジスタを取得または設定します。

処理内容

Read メソッドで `REG_ADDR.MODE2` を指定した時に MODE2 レジスタが保持されます。

Write メソッドで `REG_ADDR.MODE2` を指定した時に MODE2 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.MODE2` を指定した時に保持した値
または、SET 処理をした値

PWM0 プロパティ

概要

LED0 の輝度制御値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.PWM0` を指定した時に PWM0 レジスタが保持されます。
Write メソッドで `REG_ADDR.PWM0` を指定した時に PWM0 値を使用して書込みます。
有効値の範囲は 0~255 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.PWM0` を指定した時に保持した値
または、SET 処理をした値

PWM1 プロパティ

概要

LED1 の輝度制御値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.PWM0` を指定した時に PWM1 レジスタが保持されます。
Write メソッドで `REG_ADDR.PWM0` を指定した時に PWM1 値を使用して書込みます。
有効値の範囲は 0~255 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.PWM0` を指定した時に保持した値
または、SET 処理をした値

PWM2 プロパティ

概要

LED2 の輝度制御値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.PWM0` を指定した時に PWM2 レジスタが保持されます。
Write メソッドで `REG_ADDR.PWM0` を指定した時に PWM2 値を使用して書込みます。
有効値の範囲は 0~255 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.PWM0` を指定した時に保持した値
または、SET 処理をした値

PWM3 プロパティ

概要

LED3 の輝度制御値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.PWM0` を指定した時に PWM3 レジスタが保持されます。
Write メソッドで `REG_ADDR.PWM0` を指定した時に PWM3 値を使用して書込みます。
有効値の範囲は 0~255 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.PWM0` を指定した時に保持した値
または、SET 処理をした値

PWM4 プロパティ

概要

LED4 の輝度制御値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.PWM0` を指定した時に PWM4 レジスタが保持されます。
Write メソッドで `REG_ADDR.PWM0` を指定した時に PWM4 値を使用して書込みます。
有効値の範囲は 0~255 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.PWM0` を指定した時に保持した値
または、SET 処理をした値

PWM5 プロパティ

概要

LED5 の輝度制御値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.PWM0` を指定した時に PWM5 レジスタが保持されます。
Write メソッドで `REG_ADDR.PWM0` を指定した時に PWM5 値を使用して書込みます。
有効値の範囲は 0~255 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.PWM0` を指定した時に保持した値
または、SET 処理をした値

PWM6 プロパティ

概要

LED6 の輝度制御値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.PWM0` を指定した時に PWM6 レジスタが保持されます。
Write メソッドで `REG_ADDR.PWM0` を指定した時に PWM6 値を使用して書込みます。
有効値の範囲は 0~255 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.PWM0` を指定した時に保持した値
または、SET 処理をした値

PWM7 プロパティ

概要

LED7 の輝度制御値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.PWM0` を指定した時に PWM7 レジスタが保持されます。
Write メソッドで `REG_ADDR.PWM0` を指定した時に PWM7 値を使用して書込みます。
有効値の範囲は 0~255 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.PWM0` を指定した時に保持した値
または、SET 処理をした値

PWM8 プロパティ

概要

LED8 の輝度制御値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.PWM0` を指定した時に PWM8 レジスタが保持されます。
Write メソッドで `REG_ADDR.PWM0` を指定した時に PWM8 値を使用して書込みます。
有効値の範囲は 0~255 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.PWM0` を指定した時に保持した値
または、SET 処理をした値

PWM9 プロパティ

概要

LED9 の輝度制御値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.PWM0` を指定した時に PWM9 レジスタが保持されます。
Write メソッドで `REG_ADDR.PWM0` を指定した時に PWM9 値を使用して書込みます。
有効値の範囲は 0~255 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.PWM0` を指定した時に保持した値
または、SET 処理をした値

PWM10 プロパティ

概要

LED10 の輝度制御値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.PWM0` を指定した時に PWM10 レジスタが保持されます。
Write メソッドで `REG_ADDR.PWM0` を指定した時に PWM10 値を使用して書込みます。
有効値の範囲は 0~255 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.PWM0` を指定した時に保持した値
または、SET 処理をした値

PWM11 プロパティ

概要

LED11 の輝度制御値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.PWM0` を指定した時に PWM11 レジスタが保持されます。
Write メソッドで `REG_ADDR.PWM0` を指定した時に PWM11 値を使用して書込みます。
有効値の範囲は 0~255 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.PWM0` を指定した時に保持した値
または、SET 処理をした値

PWM12 プロパティ

概要

LED12 の輝度制御値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.PWM0` を指定した時に PWM12 レジスタが保持されます。
Write メソッドで `REG_ADDR.PWM0` を指定した時に PWM12 値を使用して書込みます。
有効値の範囲は 0~255 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.PWM0` を指定した時に保持した値
または、SET 処理をした値

PWM13 プロパティ

概要

LED13 の輝度制御値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.PWM0` を指定した時に PWM13 レジスタが保持されます。
Write メソッドで `REG_ADDR.PWM0` を指定した時に PWM13 値を使用して書込みます。
有効値の範囲は 0~255 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.PWM0` を指定した時に保持した値
または、SET 処理をした値

PWM14 プロパティ

概要

LED14 の輝度制御値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.PWM0` を指定した時に PWM14 レジスタが保持されます。
Write メソッドで `REG_ADDR.PWM0` を指定した時に PWM14 値を使用して書込みます。
有効値の範囲は 0~255 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.PWM0` を指定した時に保持した値
または、SET 処理をした値

PWM15 プロパティ

概要

LED15 の輝度制御値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.PWM0` を指定した時に PWM15 レジスタが保持されます。
Write メソッドで `REG_ADDR.PWM0` を指定した時に PWM15 値を使用して書込みます。
有効値の範囲は 0~255 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.PWM0` を指定した時に保持した値
または、SET 処理をした値

GRPPWM プロパティ

概要

グループ輝度制御レジスタを取得または設定します。

処理内容

Read メソッドで `REG_ADDR.GRPPWM` を指定した時に GRPPWM レジスタが保持されます。
Write メソッドで `REG_ADDR.GRPPWM` を指定した時に GRPPWM 値を使用して書込みます。
有効値の範囲は 0~255 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.GRPPWM` を指定した時に保持した値
または、SET 処理をした値

GRPFREQ プロパティ

概要

グループ周波数レジスタ(点滅周期)を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.GRPPWM` を指定した時に GRPFREQ レジスタが保持されます。
Write メソッドで `REG_ADDR.GRPPWM` を指定した時に GRPFREQ 値を使用して書込みます。
有効値の範囲は 0~255 です。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.GRPPWM` を指定した時に保持した値
または、SET 処理をした値

LDR0 プロパティ

概要

LED0 の出力制御を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.LEDOUT0` を指定した時に LEDOUT0 レジスタの LDR0 が保持されます。

Write メソッドで `REG_ADDR.LEDOUT0` を指定した時に LDR0 値を使用して書込みます。

0: LED OFF (Default)

1: LED ON

2: PWD レジスタによる輝度設定

3: PWD レジスタと GRPPWM レジスタによる輝度設定

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.LEDOUT0` を指定した時に保持した値
または、SET 処理をした値

LDR1 プロパティ

概要

LED1 の出力制御を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.LEDOUT0` を指定した時に LEDOUT0 レジスタの LDR1 が保持されます。

Write メソッドで `REG_ADDR.LEDOUT0` を指定した時に LDR1 値を使用して書込みます。

0: LED OFF (Default)

1: LED ON

2: PWD レジスタによる輝度設定

3: PWD レジスタと GRPPWM レジスタによる輝度設定

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.LEDOUT0` を指定した時に保持した値
または、SET 処理をした値

LDR2 プロパティ

概要

LED2 の出力制御を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.LEDOUT0` を指定した時に LEDOUT0 レジスタの LDR2 が保持されます。

Write メソッドで `REG_ADDR.LEDOUT0` を指定した時に LDR2 値を使用して書込みます。

0: LED OFF (Default)

1: LED ON

2: PWD レジスタによる輝度設定

3: PWD レジスタと GRPPWM レジスタによる輝度設定

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.LEDOUT0` を指定した時に保持した値
または、SET 処理をした値

LDR3 プロパティ

概要

LED3 の出力制御を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.LEDOUT0` を指定した時に LEDOUT0 レジスタの LDR3 が保持されます。

Write メソッドで `REG_ADDR.LEDOUT0` を指定した時に LDR3 値を使用して書込みます。

0: LED OFF (Default)

1: LED ON

2: PWD レジスタによる輝度設定

3: PWD レジスタと GRPPWM レジスタによる輝度設定

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.LEDOUT0` を指定した時に保持した値
または、SET 処理をした値

LDR4 プロパティ

概要

LED4 の出力制御を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.LEDOUT0` を指定した時に LEDOUT1 レジスタの LDR4 が保持されます。

Write メソッドで `REG_ADDR.LEDOUT0` を指定した時に LDR4 値を使用して書込みます。

0: LED OFF (Default)

1: LED ON

2: PWD レジスタによる輝度設定

3: PWD レジスタと GRPPWM レジスタによる輝度設定

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.LEDOUT0` を指定した時に保持した値
または、SET 処理をした値

LDR5 プロパティ

概要

LED5 の出力制御を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.LEDOUT0` を指定した時に LEDOUT1 レジスタの LDR5 が保持されます。

Write メソッドで `REG_ADDR.LEDOUT0` を指定した時に LDR5 値を使用して書込みます。

0: LED OFF (Default)

1: LED ON

2: PWD レジスタによる輝度設定

3: PWD レジスタと GRPPWM レジスタによる輝度設定

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.LEDOUT0` を指定した時に保持した値
または、SET 処理をした値

LDR6 プロパティ

概要

LED6 の出力制御を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.LEDOUT0` を指定した時に LEDOUT1 レジスタの LDR6 が保持されます。

Write メソッドで `REG_ADDR.LEDOUT0` を指定した時に LDR6 値を使用して書込みます。

0: LED OFF (Default)

1: LED ON

2: PWD レジスタによる輝度設定

3: PWD レジスタと GRPPWM レジスタによる輝度設定

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.LEDOUT0` を指定した時に保持した値
または、SET 処理をした値

LDR7 プロパティ

概要

LED7 の出力制御を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.LEDOUT0` を指定した時に LEDOUT1 レジスタの LDR7 が保持されます。

Write メソッドで `REG_ADDR.LEDOUT0` を指定した時に LDR7 値を使用して書込みます。

0: LED OFF (Default)

1: LED ON

2: PWD レジスタによる輝度設定

3: PWD レジスタと GRPPWM レジスタによる輝度設定

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.LEDOUT0` を指定した時に保持した値
または、SET 処理をした値

LDR8 プロパティ

概要

LED8 の出力制御を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.LEDOUT0` を指定した時に LEDOUT2 レジスタの LDR8 が保持されます。

Write メソッドで `REG_ADDR.LEDOUT0` を指定した時に LDR8 値を使用して書込みます。

0: LED OFF (Default)

1: LED ON

2: PWD レジスタによる輝度設定

3: PWD レジスタと GRPPWM レジスタによる輝度設定

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.LEDOUT0` を指定した時に保持した値
または、SET 処理をした値

LDR9 プロパティ

概要

LED9 の出力制御を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.LEDOUT0` を指定した時に LEDOUT2 レジスタの LDR9 が保持されます。

Write メソッドで `REG_ADDR.LEDOUT0` を指定した時に LDR9 値を使用して書込みます。

0: LED OFF (Default)

1: LED ON

2: PWD レジスタによる輝度設定

3: PWD レジスタと GRPPWM レジスタによる輝度設定

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.LEDOUT0` を指定した時に保持した値
または、SET 処理をした値

LDR10 プロパティ

概要

LED10 の出力制御を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.LEDOUT0` を指定した時に LEDOUT2 レジスタの LDR10 が保持されます。

Write メソッドで `REG_ADDR.LEDOUT0` を指定した時に LDR10 値を使用して書込みます。

0: LED OFF (Default)

1: LED ON

2: PWD レジスタによる輝度設定

3: PWD レジスタと GRPPWM レジスタによる輝度設定

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.LEDOUT0` を指定した時に保持した値
または、SET 処理をした値

LDR11 プロパティ

概要

LED11 の出力制御を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.LEDOUT0` を指定した時に LEDOUT2 レジスタの LDR11 が保持されます。

Write メソッドで `REG_ADDR.LEDOUT0` を指定した時に LDR11 値を使用して書込みます。

0: LED OFF (Default)

1: LED ON

2: PWD レジスタによる輝度設定

3: PWD レジスタと GRPPWM レジスタによる輝度設定

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.LEDOUT0` を指定した時に保持した値
または、SET 処理をした値

LDR12 プロパティ

概要

LED12 の出力制御を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.LEDOUT0` を指定した時に LEDOUT3 レジスタの LDR12 が保持されます。

Write メソッドで `REG_ADDR.LEDOUT0` を指定した時に LDR12 値を使用して書込みます。

0: LED OFF (Default)

1: LED ON

2: PWD レジスタによる輝度設定

3: PWD レジスタと GRPPWM レジスタによる輝度設定

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.LEDOUT0` を指定した時に保持した値
または、SET 処理をした値

LDR13 プロパティ

概要

LED13 の出力制御を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.LEDOUT0` を指定した時に LEDOUT3 レジスタの LDR13 が保持されます。

Write メソッドで `REG_ADDR.LEDOUT0` を指定した時に LDR13 値を使用して書込みます。

0: LED OFF (Default)

1: LED ON

2: PWD レジスタによる輝度設定

3: PWD レジスタと GRPPWM レジスタによる輝度設定

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.LEDOUT0` を指定した時に保持した値
または、SET 処理をした値

LDR14 プロパティ

概要

LED14 の出力制御を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.LEDOUT0` を指定した時に LEDOUT3 レジスタの LDR14 が保持されます。

Write メソッドで `REG_ADDR.LEDOUT0` を指定した時に LDR14 値を使用して書込みます。

0: LED OFF (Default)

1: LED ON

2: PWD レジスタによる輝度設定

3: PWD レジスタと GRPPWM レジスタによる輝度設定

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.LEDOUT0` を指定した時に保持した値
または、SET 処理をした値

LDR15 プロパティ

概要

LED15 の出力制御を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.LEDOUT0` を指定した時に LEDOUT3 レジスタの LDR15 が保持されます。

Write メソッドで `REG_ADDR.LEDOUT0` を指定した時に LDR15 値を使用して書込みます。

0: LED OFF (Default)

1: LED ON

2: PWD レジスタによる輝度設定

3: PWD レジスタと GRPPWM レジスタによる輝度設定

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.LEDOUT0` を指定した時に保持した値
または、SET 処理をした値

SUBADR1 プロパティ

概要

I2C-bus subaddress 1 を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.SUBADR1` を指定した時に SUBADR1 レジスタの値が保持されます。

Write メソッドで `REG_ADDR.SUBADR1` を指定した時に SUBADR1 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.SUBADR1` を指定した時に保持した値
または、SET 処理をした値

SUBADR2 プロパティ

概要

I2C-bus subaddress 2 を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.SUBADR1` を指定した時に SUBADR2 レジスタの値が保持されます。

Write メソッドで `REG_ADDR.SUBADR1` を指定した時に SUBADR2 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.SUBADR1` を指定した時に保持した値
または、SET 処理をした値

SUBADR3 プロパティ

概要

I2C-bus subaddress 3 を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.SUBADR1` を指定した時に SUBADR3 レジスタの値が保持されます。

Write メソッドで `REG_ADDR.SUBADR1` を指定した時に SUBADR3 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.SUBADR1` を指定した時に保持した値
または、SET 処理をした値

ALLCALLADR プロパティ

概要

ALLCALL I2C-bus address register を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.ALLCALLADR` を指定した時に ALLCALLADR レジスタの値が保持されます。

Write メソッドで `REG_ADDR.ALLCALLADR` を指定した時に ALLCALLADR 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.ALLCALLADR` を指定した時に保持した値
または、SET 処理をした値

Init メソッド

概要

I2C バスコントローラに指定された接続設定を使って非同期的に接続します。

引数

| 引数名 | 型 | 説明 |
|---------|------|-----------|
| Address | byte | I2C のアドレス |

戻り値

なし

Read メソッド

概要

最新の変換結果を読み込みます。

引数

| 引数名 | 型 | 説明 |
|----------|------|--|
| REG_ADDR | byte | REG_ADDR_MODE1 (0x00) ~ REG_ADDR_ALLCALLADR (0x1B) |

戻り値

Status と同じ

1: 正常

1 以外: 異常

読込処理

| REG_ADDR 値 | 読込プロパティ |
|------------|--|
| MODE1 | MODE1 |
| MODE2 | MODE2 |
| PWM0 | PWM0 PWM1 PWM2 PWM3 PWM4 PWM5 PWM6 PWM7 PWM8 PWM9 PWM10 PWM11 PWM12 PWM13 PWM14 PWM15 |
| GRPPWM | GRPPWM GRPFREQ |

| REG_ADDR 値 | 読込プロパティ |
|------------|--|
| LEDOUT0 | LDR0 LDR1 LDR2 LDR3 LDR4 LDR5 LDR6 LDR7 LDR8 LDR9 LDR10 LDR11 LDR12 LDR13 LDR14 LDR15 |
| SUBADR1 | SUBADR1 SUBADR2 SUBADR3 |
| ALLCALLADR | ALLCALLADR |

Write メソッド

概要

引数 REG_ADDR に関するプロパティを編集し書き込みます。

引数

| 引数名 | 型 | 説明 |
|----------|------|--|
| REG_ADDR | byte | REG_ADDR_MODE1 (0x00) ~ REG_ADDR_ALLCALLADR (0x1B) |

戻り値

Status

1: 正常

1 以外: 異常

書込処理

| REG_ADDR 値 | 書込プロパティ |
|------------|--|
| MODE1 | MODE1 |
| MODE2 | MODE2 |
| PWM0 | PWM0 PWM1 PWM2 PWM3 PWM4 PWM5 PWM6 PWM7 PWM8 PWM9 PWM10 PWM11 PWM12 PWM13 PWM14 PWM15 |
| GRPPWM | GRPPWM GRPFREQ |

| REG_ADDR 値 | 読込プロパティ |
|------------|--|
| LEDOUT0 | LDR0 LDR1 LDR2 LDR3 LDR4 LDR5 LDR6 LDR7 LDR8 LDR9 LDR10 LDR11 LDR12 LDR13 LDR14 LDR15 |
| SUBADR1 | SUBADR1 SUBADR2 SUBADR3 |
| ALLCALLADR | ALLCALLADR |

サンプルプログラム

```
Imports I2CLibrary
Public NotInheritable Class PCA9622Page
    Inherits Page

    Private Const PCA9622_Address As Byte = &H50
    Private PCA9622 As PCA9622 = New PCA9622()

    Public Sub New()
        InitializeComponent()
        Call Init()
    End Sub

    Private Async Sub Init()
        Await PCA9622.Init(PCA9622_Address)
    End Sub

    Private Sub btnLEDOn_Click(sender As Object, e As RoutedEventArgs)
        PCA9622.Read(PCA9622.REG_ADDR.MODE1)
        PCA9622.Read(PCA9622.REG_ADDR.MODE2)
        PCA9622.Read(PCA9622.REG_ADDR.PWM0)
        PCA9622.Read(PCA9622.REG_ADDR.LEDOUT0)

        PCA9622.MODE1 = &H0
        PCA9622.Write(PCA9622.REG_ADDR.MODE1)

        PCA9622.LDR0 = &H0
        PCA9622.LDR1 = &H0
        PCA9622.LDR2 = &H0
        PCA9622.LDR3 = &H2
        PCA9622.LDR4 = &H2
        PCA9622.LDR5 = &H0
        PCA9622.LDR6 = &H0
        PCA9622.LDR7 = &H0
        PCA9622.LDR8 = &H2
        PCA9622.LDR9 = &H2
        PCA9622.LDR10 = &H2
        PCA9622.LDR11 = &H2
        PCA9622.LDR12 = &H2
        PCA9622.LDR13 = &H2
        PCA9622.LDR14 = &H2
        PCA9622.LDR15 = &H2

        PCA9622.Write(PCA9622.REG_ADDR.LEDOUT0)

        PCA9622.PWM0 = &H0
        PCA9622.PWM1 = &H0
        PCA9622.PWM2 = &H0
        PCA9622.PWM3 = &HFF
        PCA9622.PWM4 = &HFF
        PCA9622.PWM5 = &H0
    End Sub
End Class
```

```
PCA9622.PWM6 = &H0
PCA9622.PWM7 = &H0
PCA9622.PWM8 = &H2
PCA9622.PWM9 = &H4
PCA9622.PWM10 = &H8
PCA9622.PWM11 = &H10
PCA9622.PWM12 = &H20
PCA9622.PWM13 = &H40
PCA9622.PWM14 = &H80
PCA9622.PWM15 = &HFF
```

```
PCA9622.Write(PCA9622.REG_ADDR.PWM0)
```

End Sub

```
Private Sub btnLEDOff_Click(sender As Object, e As RoutedEventArgs)
```

```
PCA9622.LDR0 = 0
PCA9622.LDR1 = 0
PCA9622.LDR2 = 0
PCA9622.LDR3 = 0
PCA9622.LDR4 = 0
PCA9622.LDR5 = 0
PCA9622.LDR6 = 0
PCA9622.LDR7 = 0
PCA9622.LDR8 = 0
PCA9622.LDR9 = 0
PCA9622.LDR10 = 0
PCA9622.LDR11 = 0
PCA9622.LDR12 = 0
PCA9622.LDR13 = 0
PCA9622.LDR14 = 0
PCA9622.LDR15 = 0
```

```
PCA9622.Write(PCA9622.REG_ADDR.LEDOUT0)
```

End Sub

```
Private Sub btnBack_Click(sender As Object, e As RoutedEventArgs)
```

```
RefreshTimer.Stop()
ADC121C021.close()
App.rootFrame.Navigate(GetType(MenuPage))
```

End Sub

End Class

3-12.3 軸加速度センサ ADXL345

変更履歴

2016年11月21日 第一版 新規作成

1. 概要

3軸加速度センサーとI2C 接続するための機能を提供します。

2. 機能一覧

プロパティ

| 機能名 | 機能説明 |
|----------------|------------------------------|
| DEVID | Device ID を取得します。 |
| THRESH_TAP | タップしきい値を取得または設定します。 |
| OFSX | X-axis オフセットを取得または設定します。 |
| OFSY | Y-axis オフセットを取得または設定します。 |
| OFSZ | Z-axis オフセットを取得または設定します。 |
| DUR | タップ持続時間を取得または設定します。 |
| Latent | タップ待ち時間を取得または設定します。 |
| Window | タップ Window を取得または設定します。 |
| THRESH_ACT | Activity しきい値を取得または設定します。 |
| THRESH_INACT | Inactivity しきい値を取得または設定します。 |
| TIME_INACT | Inactivity time を取得または設定します。 |
| INACT_Z_enable | INACT_Z_enable を取得または設定します。 |
| INACT_Y_enable | INACT_Y_enable を取得または設定します。 |

| 機能名 | 機能説明 |
|----------------|-----------------------------|
| INACT_X_enable | INACT_X_enable を取得または設定します。 |
| INACT_acdc | INACT_acdc を取得または設定します。 |
| ACT_Z_enable | ACT_Z_enable を取得または設定します。 |
| ACT_Y_enable | ACT_Y_enable を取得または設定します。 |
| ACT_X_enable | ACT_X_enable を取得または設定します。 |
| ACT_acdc | ACT_acdc を取得または設定します。 |
| THRESH_FF | Free-fall しきい値を取得または設定します。 |
| TIME_FF | Free-fall time を取得または設定します。 |
| TAP_Z_enable | TAP_Z_enable を取得または設定します。 |
| TAP_Y_enable | TAP_Y_enable を取得または設定します。 |
| TAP_X_enable | TAP_X_enable を取得または設定します。 |
| Suppress | Suppress を取得または設定します。 |
| TAP_Z_source | TAP_Z_source を取得します。 |
| TAP_Y_source | TAP_Y_source を取得します。 |
| TAP_X_source | TAP_X_source を取得します。 |
| Asleep | Asleep を取得します。 |
| ACT_Z_source | ACT_Z_source を取得します。 |
| ACT_Y_source | ACT_Y_source を取得します。 |
| ACT_X_source | ACT_X_source を取得します。 |
| Rate | Rate を取得または設定します。 |

| 機能名 | 機能説明 |
|-----------------------|--------------------------------------|
| LOW_POWER | LOW_POWER を取得または設定します。 |
| Wakeup | Wakeup を取得または設定します。 |
| Sleep | Sleep を取得または設定します。 |
| Measure | Measure を取得または設定します。 |
| AUTO_SLEEP | AUTO_SLEEP を取得または設定します。 |
| Link | Link を取得または設定します。 |
| INT_ENABLE_Overrun | INT_ENABLE の Overrun を取得または設定します。 |
| INT_ENABLE_Watermark | INT_ENABLE の Watermark を取得または設定します。 |
| INT_ENABLE_FREE_FALL | INT_ENABLE の FREE_FALL を取得または設定します。 |
| INT_ENABLE_Inactivity | INT_ENABLE の Inactivity を取得または設定します。 |
| INT_ENABLE_Activity | INT_ENABLE の Activity を取得または設定します。 |
| INT_ENABLE_DOUBLE_TAP | INT_ENABLE の DOUBLE_TAP を取得または設定します。 |
| INT_ENABLE_SINGLE_TAP | INT_ENABLE の SINGLE_TAP を取得または設定します。 |
| INT_ENABLE_DATA_READY | INT_ENABLE の DATA_READY を取得または設定します。 |
| INT_MAP_Overrun | INT_MAP の Overrun を取得または設定します。 |
| INT_MAP_Watermark | INT_MAP の Watermark を取得または設定します。 |
| INT_MAP_FREE_FALL | INT_MAP の FREE_FALL を取得または設定します。 |
| INT_MAP_Inactivity | INT_MAP の Inactivity を取得または設定します。 |
| INT_MAP_Activity | INT_MAP の Activity を取得または設定します。 |
| INT_MAP_DOUBLE_TAP | INT_MAP の DOUBLE_TAP を取得または設定します。 |

| 機能名 | 機能説明 |
|-----------------------|-----------------------------------|
| INT_MAP_SINGLE_TAP | INT_MAP の SINGLE_TAP を取得または設定します。 |
| INT_MAP_DATA_READY | INT_MAP の DATA_READY を取得または設定します。 |
| INT_SOURCE_Overrun | INT_SOURCE の Overrun を取得します。 |
| INT_SOURCE_Watermark | INT_SOURCE の Watermark を取得します。 |
| INT_SOURCE_FREE_FALL | INT_SOURCE の FREE_FALL を取得します。 |
| INT_SOURCE_Inactivity | INT_SOURCE の Inactivity を取得します。 |
| INT_SOURCE_Activity | INT_SOURCE の Activity を取得します。 |
| INT_SOURCE_DOUBLE_TAP | INT_SOURCE の DOUBLE_TAP を取得します。 |
| INT_SOURCE_SINGLE_TAP | INT_SOURCE の SINGLE_TAP を取得します。 |
| INT_SOURCE_DATA_READY | INT_SOURCE の DATA_READY を取得します。 |
| Range | Range を取得または設定します。 |
| Justify | Justify を取得または設定します。 |
| FULL_RES | FULL_RES を取得または設定します。 |
| INT_INVERT | INT_INVERT を取得または設定します。 |
| SPI | SPI を取得または設定します。 |
| SELF_TEST | SELF_TEST を取得または設定します。 |
| DATA_X | X-Axis Data を取得します。 |
| DATA_Y | Y-Axis Data を取得します。 |
| DATA_Z | Z-Axis Data を取得します。 |
| Samples | Samples を取得または設定します。 |
| Trigger | Trigger を取得または設定します。 |
| FIFO_MODE | FIFO_MODE を取得または設定します。 |
| Entries | Entries を取得します。 |
| FIFO_TRIG | FIFO_TRIG を取得します。 |

メソッド

| 機能名 | 機能説明 |
|-------|---------------------------------------|
| Init | I2C バスコントローラに指定された接続設定を使って非同期的に接続します。 |
| Read | 指定レジスタ読込 |
| Write | 指定レジスタ書込 |

定数

| 機能名 | 機能説明 | 定数値 |
|-------------------------|---------------------|------|
| REG_ADDR.DEVID | レジスタ DEVID | 0x00 |
| REG_ADDR.THRESH_TAP | レジスタ THRESH_TAP | 0x1D |
| REG_ADDR.OFSX | レジスタ OFSX | 0x1E |
| REG_ADDR.DUR | レジスタ DUR | 0x21 |
| REG_ADDR.Latent | レジスタ Latent | 0x22 |
| REG_ADDR.Window | レジスタ Window | 0x23 |
| REG_ADDR.THRESH_ACT | レジスタ THRESH_ACT | 0x24 |
| REG_ADDR.THRESH_INACT | レジスタ THRESH_INACT | 0x25 |
| REG_ADDR.TIME_INACT | レジスタ TIME_INACT | 0x26 |
| REG_ADDR.ACT_INACT_CTL | レジスタ ACT_INACT_CTL | 0x27 |
| REG_ADDR.THRESH_FF | レジスタ THRESH_FF | 0x28 |
| REG_ADDR.TIME_FF | レジスタ TIME_FF | 0x29 |
| REG_ADDR.TAP_AXES | レジスタ TAP_AXES | 0x2A |
| REG_ADDR.ACT_TAP_STATUS | レジスタ ACT_TAP_STATUS | 0x2B |
| REG_ADDR.BW_RATE | レジスタ BW_RATE | 0x2C |
| REG_ADDR.POWER_CTL | レジスタ POWER_CTL | 0x2D |
| REG_ADDR.INT_ENABLE | レジスタ INT_ENABLE | 0x2E |
| REG_ADDR.INT_MAP | レジスタ INT_MAP | 0x2F |
| REG_ADDR.INT_SOURCE | レジスタ INT_SOURCE | 0x30 |
| REG_ADDR.DATA_FORMAT | レジスタ DATA_FORMAT | 0x31 |
| REG_ADDR.DATAX0 | レジスタ DATAX0 | 0x32 |
| REG_ADDR.FIFO_CTL | レジスタ FIFO_CTL | 0x38 |
| REG_ADDR.FIFO_STATUS | レジスタ FIFO_STATUS | 0x39 |

DEVID プロパティ

概要

Device ID を取得します。

処理内容

Read メソッドで `REG_ADDR.DEVID` を指定した時に DEVID にレジスタが保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.DEVID` を指定した時に保持した値
または、SET 処理をした値

THRESH_TAP プロパティ

概要

タップしきい値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.THRESH_TAP` を指定した時に `THRESH_TAP` レジスタが保持されます。

Write メソッドで `REG_ADDR.THRESH_TAP` を指定した時に `THRESH_TAP` 値を使用して書込みます。

SET 処理

`value` 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.THRESH_TAP` を指定した時に保持した値
または、SET 処理をした値

OFSX プロパティ

概要

X-axis オフセットを取得または設定します。

処理内容

Read メソッドで `REG_ADDR.OFSX` を指定した時に OFSX レジスタが保持されます。

Write メソッドで `REG_ADDR.OFSX` を指定した時に OFSX 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.OFSX` を指定した時に保持した値
または、SET 処理をした値

OFSY プロパティ

概要

Y-axis オフセットを取得または設定します。

処理内容

Read メソッドで `REG_ADDR.OFSX` を指定した時に OFSY レジスタが保持されます。

Write メソッドで `REG_ADDR.OFSX` を指定した時に OFSY 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.OFSX` を指定した時に保持した値
または、SET 処理をした値

OFSZ プロパティ

概要

Z-axis オフセットを取得または設定します。

処理内容

Read メソッドで `REG_ADDR.OFSX` を指定した時に OFSZ レジスタが保持されます。

Write メソッドで `REG_ADDR.OFSX` を指定した時に OFSZ 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.OFSX` を指定した時に保持した値
または、SET 処理をした値

DUR プロパティ

概要

タップ持続時間を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.DUR` を指定した時に DUR レジスタが保持されます。

Write メソッドで `REG_ADDR.DUR` を指定した時に DUR 値を使用して書込ます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.DUR` を指定した時に保持した値
または、SET 処理をした値

Latent プロパティ

概要

タップ待ち時間を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Latent` を指定した時に Latent レジスタが保持されます。

Write メソッドで `REG_ADDR.Latent` を指定した時に Latent 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Latent` を指定した時に保持した値
または、SET 処理をした値

Window プロパティ

概要

タップ Window を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.Window` を指定した時に Window レジスタが保持されます。

Write メソッドで `REG_ADDR.Window` を指定した時に Window 値を使用して書込ます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.Window` を指定した時に保持した値
または、SET 処理をした値

THRESH_ACT プロパティ

概要

Activity しきい値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.THRESH_ACT` を指定した時に `THRESH_ACT` レジスタが保持されます。

Write メソッドで `REG_ADDR.THRESH_ACT` を指定した時に `THRESH_ACT` 値を使用して書込みます。

SET 処理

`value` 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.THRESH_ACT` を指定した時に保持した値
または、SET 処理をした値

THRESH_INACT プロパティ

概要

Inactivity しきい値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.THRESH_INACT` を指定した時に `THRESH_INACT` レジスタが保持されます。

Write メソッドで `REG_ADDR.THRESH_INACT` を指定した時に `THRESH_INACT` 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.THRESH_INACT` を指定した時に保持した値
または、SET 処理をした値

TIME_INACT プロパティ

概要

Inactivity time を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.TIME_INACT` を指定した時に `TIME_INACT` レジスタが保持されます。

Write メソッドで `REG_ADDR.TIME_INACT` を指定した時に `TIME_INACT` 値を使用して書込まれます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.TIME_INACT` を指定した時に保持した値
または、SET 処理をした値

INACT_Z_enable プロパティ

概要

INACT_Z_enable を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.ACT_INACT_CTL` を指定した時に `ACT_INACT_CTL` レジスタの `INACT_Z_enable` が保持されます。

Write メソッドで `REG_ADDR.ACT_INACT_CTL` を指定した時に `INACT_Z_enable` 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.ACT_INACT_CTL` を指定した時に保持した値
または、SET 処理をした値

INACT_Y_enable プロパティ

概要

INACT_Y_enable を取得または設定します。

処理内容

Read メソッドで [REG_ADDR.ACT_INACT_CTL](#) を指定した時に ACT_INACT_CTL レジスタの INACT_Y_enable が保持されます。

Write メソッドで [REG_ADDR.ACT_INACT_CTL](#) を指定した時に INACT_Y_enable 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで [REG_ADDR.ACT_INACT_CTL](#) を指定した時に保持した値
または、SET 処理をした値

INACT_X_enable プロパティ

概要

INACT_X_enable を取得または設定します。

処理内容

Read メソッドで [REG_ADDR.ACT_INACT_CTL](#) を指定した時に ACT_INACT_CTL レジスタの INACT_X_enable が保持されます。

Write メソッドで [REG_ADDR.ACT_INACT_CTL](#) を指定した時に INACT_X_enable 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで [REG_ADDR.ACT_INACT_CTL](#) を指定した時に保持した値
または、SET 処理をした値

INACT_acdc プロパティ

概要

INACT_acdc を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.ACT_INACT_CTL` を指定した時に `ACT_INACT_CTL` レジスタの `INACT_acdc` が保持されます。

Write メソッドで `REG_ADDR.ACT_INACT_CTL` を指定した時に `INACT_acdc` 値を使用して書込みます。

SET 処理

`value` 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.ACT_INACT_CTL` を指定した時に保持した値
または、SET 処理をした値

ACT_Z_enable プロパティ

概要

ACT_Z_enable を取得または設定します。

処理内容

Read メソッドで [REG_ADDR.ACT_INACT_CTL](#) を指定した時に ACT_INACT_CTL レジスタの ACT_Z_enable が保持されます。

Write メソッドで [REG_ADDR.ACT_INACT_CTL](#) を指定した時に ACT_Z_enable 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで [REG_ADDR.ACT_INACT_CTL](#) を指定した時に保持した値または、SET 処理をした値

ACT_Y_enable プロパティ

概要

ACT_Y_enable を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.ACT_INACT_CTL` を指定した時に `ACT_INACT_CTL` レジスタの `ACT_Y_enable` が保持されます。

Write メソッドで `REG_ADDR.ACT_INACT_CTL` を指定した時に `ACT_Y_enable` 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.ACT_INACT_CTL` を指定した時に保持した値
または、SET 処理をした値

ACT_X_enable プロパティ

概要

ACT_X_enable を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.ACT_INACT_CTL` を指定した時に `ACT_INACT_CTL` レジスタの `ACT_X_enable` が保持されます。

Write メソッドで `REG_ADDR.ACT_INACT_CTL` を指定した時に `ACT_X_enable` 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.ACT_INACT_CTL` を指定した時に保持した値
または、SET 処理をした値

ACT_acdc プロパティ

概要

ACT_acdc を取得または設定します。

処理内容

Read メソッドで [REG_ADDR.ACT_INACT_CTL](#) を指定した時に ACT_INACT_CTL レジスタの ACT_acdc が保持されます。

Write メソッドで [REG_ADDR.ACT_INACT_CTL](#) を指定した時に ACT_acdc 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで [REG_ADDR.ACT_INACT_CTL](#) を指定した時に保持した値
または、SET 処理をした値

THRESH_FF プロパティ

概要

Free-fall しきい値を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.THRESH_FF` を指定した時に `THRESH_FF` レジスタが保持されます。

Write メソッドで `REG_ADDR.THRESH_FF` を指定した時に `THRESH_FF` 値を使用して書込みます。

SET 処理

`value` 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.THRESH_FF` を指定した時に保持した値
または、SET 処理をした値

TIME_FF プロパティ

概要

Free-fall time を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.TIME_FF` を指定した時に `TIME_FF` レジスタが保持されます。

Write メソッドで `REG_ADDR.TIME_FF` を指定した時に `TIME_FF` 値を使用して書込みます。

SET 処理

`value` 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.TIME_FF` を指定した時に保持した値
または、SET 処理をした値

TAP_Z_enable プロパティ

概要

TAP_Z_enable を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.TAP_AXES` を指定した時に TAP_AXES レジスタの TAP_Z_enable が保持されます。

Write メソッドで `REG_ADDR.TAP_AXES` を指定した時に TAP_Z_enable 値を使用して書込ます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.TAP_AXES` を指定した時に保持した値
または、SET 処理をした値

TAP_Y_enable プロパティ

概要

TAP_Y_enable を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.TAP_AXES` を指定した時に TAP_AXES レジスタの TAP_Y_enable が保持されます。

Write メソッドで `REG_ADDR.TAP_AXES` を指定した時に TAP_Y_enable 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.TAP_AXES` を指定した時に保持した値
または、SET 処理をした値

TAP_X_enable プロパティ

概要

TAP_X_enable を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.TAP_AXES` を指定した時に TAP_AXES レジスタの TAP_X_enable が保持されます。

Write メソッドで `REG_ADDR.TAP_AXES` を指定した時に TAP_X_enable 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.TAP_AXES` を指定した時に保持した値
または、SET 処理をした値

Suppress プロパティ

概要

Suppress を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.TAP_AXES` を指定した時に `TAP_AXES` レジスタの Suppress が保持されます。

Write メソッドで `REG_ADDR.TAP_AXES` を指定した時に Suppress 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.TAP_AXES` を指定した時に保持した値
または、SET 処理をした値

TAP_Z_source プロパティ

概要

TAP_Z_source を取得します。

処理内容

Read メソッドで `REG_ADDR.ACT_TAP_STATUS` を指定した時に `ACT_TAP_STATUS` レジスタの TAP_Z_source が保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.ACT_TAP_STATUS` を指定した時に保持した値または、SET 処理をした値

TAP_Y_source プロパティ

概要

TAP_Y_source を取得します。

処理内容

Read メソッドで `REG_ADDR.ACT_TAP_STATUS` を指定した時に `ACT_TAP_STATUS` レジスタの TAP_Y_source が保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.ACT_TAP_STATUS` を指定した時に保持した値または、SET 処理をした値

TAP_X_source プロパティ

概要

TAP_X_source を取得します。

処理内容

Read メソッドで `REG_ADDR.ACT_TAP_STATUS` を指定した時に `ACT_TAP_STATUS` レジスタの TAP_X_source が保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.ACT_TAP_STATUS` を指定した時に保持した値または、SET 処理をした値

Asleep プロパティ

概要

Asleep を取得します。

処理内容

Read メソッドで `REG_ADDR.ACT_TAP_STATUS` を指定した時に `ACT_TAP_STATUS` レジスタの Asleep が保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.ACT_TAP_STATUS` を指定した時に保持した値または、SET 処理をした値

ACT_Z_source プロパティ

概要

ACT_Z_source を取得します。

処理内容

Read メソッドで `REG_ADDR.ACT_TAP_STATUS` を指定した時に `ACT_TAP_STATUS` レジスタの `ACT_Z_source` が保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.ACT_TAP_STATUS` を指定した時に保持した値または、SET 処理をした値

ACT_Y_source プロパティ

概要

ACT_Y_source を取得します。

処理内容

Read メソッドで `REG_ADDR.ACT_TAP_STATUS` を指定した時に `ACT_TAP_STATUS` レジスタの `ACT_Y_source` が保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.ACT_TAP_STATUS` を指定した時に保持した値または、SET 処理をした値

ACT_X_source プロパティ

概要

ACT_X_source を取得します。

処理内容

Read メソッドで `REG_ADDR.ACT_TAP_STATUS` を指定した時に `ACT_TAP_STATUS` レジスタの `ACT_X_source` が保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.ACT_TAP_STATUS` を指定した時に保持した値または、SET 処理をした値

Rate プロパティ

概要

Rate を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.BW_RATE` を指定した時に `BW_RATE` レジスタの Rate が保持されます。

Write メソッドで `REG_ADDR.BW_RATE` を指定した時に Rate 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.BW_RATE` を指定した時に保持した値
または、SET 処理をした値

LOW_POWER プロパティ

概要

LOW_POWER を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.BW_RATE` を指定した時に `BW_RATE` レジスタの `LOW_POWER` が保持されます。

Write メソッドで `REG_ADDR.BW_RATE` を指定した時に `LOW_POWER` 値を使用して書込みます。

SET 処理

`value` 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.BW_RATE` を指定した時に保持した値
または、SET 処理をした値

Wakeup プロパティ

概要

Wakeup を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.POWER_CTL` を指定した時に `POWER_CTL` レジスタの Wakeup が保持されます。

Write メソッドで `REG_ADDR.POWER_CTL` を指定した時に Wakeup 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.POWER_CTL` を指定した時に保持した値または、SET 処理をした値

Sleep プロパティ

概要

Sleep を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.POWER_CTL` を指定した時に `POWER_CTL` レジスタの Sleep が保持されます。

Write メソッドで `REG_ADDR.POWER_CTL` を指定した時に Sleep 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.POWER_CTL` を指定した時に保持した値または、SET 処理をした値

Measure プロパティ

概要

Sleep を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.POWER_CTL` を指定した時に `POWER_CTL` レジスタの Measure が保持されます。

Write メソッドで `REG_ADDR.POWER_CTL` を指定した時に Measure 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.POWER_CTL` を指定した時に保持した値または、SET 処理をした値

AUTO_SLEEP プロパティ

概要

Sleep を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.POWER_CTL` を指定した時に `POWER_CTL` レジスタの `AUTO_SLEEP` が保持されます。

Write メソッドで `REG_ADDR.POWER_CTL` を指定した時に `AUTO_SLEEP` 値を使用して書込みます。

SET 処理

`value` 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.POWER_CTL` を指定した時に保持した値
または、SET 処理をした値

Link プロパティ

概要

Link を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.POWER_CTL` を指定した時に `POWER_CTL` レジスタの Link が保持されます。

Write メソッドで `REG_ADDR.POWER_CTL` を指定した時に Link 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.POWER_CTL` を指定した時に保持した値
または、SET 処理をした値

INT_ENABLE_Overrun プロパティ

概要

INT_ENABLE の Overrun を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.INT_ENABLE` を指定した時に INT_ENABLE レジスタの Overrun が保持されます。

Write メソッドで `REG_ADDR.INT_ENABLE` を指定した時に Overrun 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.INT_ENABLE` を指定した時に保持した値
または、SET 処理をした値

INT_ENABLE_ Watermark プロパティ

概要

INT_ENABLE の Watermark を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.INT_ENABLE` を指定した時に INT_ENABLE レジスタの Watermark が保持されます。

Write メソッドで `REG_ADDR.INT_ENABLE` を指定した時に Watermark 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.INT_ENABLE` を指定した時に保持した値
または、SET 処理をした値

INT_ENABLE_ FREE_FALL プロパティ

概要

INT_ENABLE の FREE_FALL を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.INT_ENABLE` を指定した時に INT_ENABLE レジスタの FREE_FALL が保持されます。

Write メソッドで `REG_ADDR.INT_ENABLE` を指定した時に FREE_FALL 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.INT_ENABLE` を指定した時に保持した値
または、SET 処理をした値

INT_ENABLE_Inactivity プロパティ

概要

INT_ENABLE の Inactivity を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.INT_ENABLE` を指定した時に INT_ENABLE レジスタの Inactivity が保持されます。

Write メソッドで `REG_ADDR.INT_ENABLE` を指定した時に Inactivity 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.INT_ENABLE` を指定した時に保持した値
または、SET 処理をした値

INT_ENABLE_Activity プロパティ

概要

INT_ENABLE の Activity を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.INT_ENABLE` を指定した時に INT_ENABLE レジスタの Activity が保持されます。

Write メソッドで `REG_ADDR.INT_ENABLE` を指定した時に Activity 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.INT_ENABLE` を指定した時に保持した値または、SET 処理をした値

INT_ENABLE_ DOUBLE_TAP プロパティ

概要

INT_ENABLE の DOUBLE_TAP を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.INT_ENABLE` を指定した時に INT_ENABLE レジスタの DOUBLE_TAP が保持されます。

Write メソッドで `REG_ADDR.INT_ENABLE` を指定した時に DOUBLE_TAP 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.INT_ENABLE` を指定した時に保持した値
または、SET 処理をした値

INT_ENABLE_SINGLE_TAP プロパティ

概要

INT_ENABLE の SINGLE_TAP を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.INT_ENABLE` を指定した時に INT_ENABLE レジスタの SINGLE_TAP が保持されます。

Write メソッドで `REG_ADDR.INT_ENABLE` を指定した時に SINGLE_TAP 値を使用して書込ます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.INT_ENABLE` を指定した時に保持した値
または、SET 処理をした値

INT_ENABLE_ DATA_READY プロパティ

概要

INT_ENABLE の DATA_READY を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.INT_ENABLE` を指定した時に INT_ENABLE レジスタの DATA_READY が保持されます。

Write メソッドで `REG_ADDR.INT_ENABLE` を指定した時に DATA_READY 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.INT_ENABLE` を指定した時に保持した値
または、SET 処理をした値

INT_MAP_Overrun プロパティ

概要

INT_MAP の Overrun を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.INT_MAP` を指定した時に INT_MAP レジスタの Overrun が保持されます。

Write メソッドで `REG_ADDR.INT_MAP` を指定した時に INT_MAP 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.INT_MAP` を指定した時に保持した値
または、SET 処理をした値

INT_MAP_ Watermark プロパティ

概要

INT_MAP の Watermark を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.INT_MAP` を指定した時に INT_MAP レジスタの Watermark が保持されます。

Write メソッドで `REG_ADDR.INT_MAP` を指定した時に INT_MAP 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.INT_MAP` を指定した時に保持した値
または、SET 処理をした値

INT_MAP_FREE_FALL プロパティ

概要

INT_MAP の FREE_FALL を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.INT_MAP` を指定した時に INT_MAP レジスタの FREE_FALL が保持されます。

Write メソッドで `REG_ADDR.INT_MAP` を指定した時に INT_MAP 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.INT_MAP` を指定した時に保持した値
または、SET 処理をした値

INT_MAP_Inactivity プロパティ

概要

INT_MAP の Inactivity を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.INT_MAP` を指定した時に INT_MAP レジスタの Inactivity が保持されます。

Write メソッドで `REG_ADDR.INT_MAP` を指定した時に INT_MAP 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.INT_MAP` を指定した時に保持した値
または、SET 処理をした値

INT_MAP_Activity プロパティ

概要

INT_MAP の Activity を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.INT_MAP` を指定した時に INT_MAP レジスタの Activity が保持されます。

Write メソッドで `REG_ADDR.INT_MAP` を指定した時に INT_MAP 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.INT_MAP` を指定した時に保持した値
または、SET 処理をした値

INT_MAP_DOUBLE_TAP プロパティ

概要

INT_MAP の DOUBLE_TAP を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.INT_MAP` を指定した時に INT_MAP レジスタの DOUBLE_TAP が保持されます。

Write メソッドで `REG_ADDR.INT_MAP` を指定した時に INT_MAP 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.INT_MAP` を指定した時に保持した値
または、SET 処理をした値

INT_MAP_SINGLE_TAP プロパティ

概要

INT_MAP の SINGLE_TAP を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.INT_MAP` を指定した時に INT_MAP レジスタの SINGLE_TAP が保持されます。

Write メソッドで `REG_ADDR.INT_MAP` を指定した時に INT_MAP 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.INT_MAP` を指定した時に保持した値
または、SET 処理をした値

INT_MAP_DATA_READY プロパティ

概要

INT_MAP の DATA_READY を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.INT_MAP` を指定した時に INT_MAP レジスタの DATA_READY が保持されます。

Write メソッドで `REG_ADDR.INT_MAP` を指定した時に INT_MAP 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.INT_MAP` を指定した時に保持した値
または、SET 処理をした値

INT_SOURCE_Overrun プロパティ

概要

INT_SOURCE の Overrun を取得します。

処理内容

Read メソッドで `REG_ADDR.INT_SOURCE` を指定した時に INT_SOURCE レジスタの Overrun が保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.INT_SOURCE` を指定した時に保持した値
または、SET 処理をした値

INT_SOURCE_ Watermark プロパティ

概要

INT_SOURCE の Watermark を取得します。

処理内容

Read メソッドで `REG_ADDR.INT_SOURCE` を指定した時に INT_SOURCE レジスタの Watermark が保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.INT_SOURCE` を指定した時に保持した値
または、SET 処理をした値

INT_SOURCE_FREE_FALL プロパティ

概要

INT_SOURCE の FREE_FALL を取得します。

処理内容

Read メソッドで `REG_ADDR.INT_SOURCE` を指定した時に INT_SOURCE レジスタの FREE_FALL が保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.INT_SOURCE` を指定した時に保持した値
または、SET 処理をした値

INT_SOURCE_Inactivity プロパティ

概要

INT_SOURCE の Inactivity を取得します。

処理内容

Read メソッドで `REG_ADDR.INT_SOURCE` を指定した時に INT_SOURCE レジスタの Inactivity が保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.INT_SOURCE` を指定した時に保持した値
または、SET 処理をした値

INT_SOURCE_Activity プロパティ

概要

INT_SOURCE の Activity を取得します。

処理内容

Read メソッドで `REG_ADDR.INT_SOURCE` を指定した時に INT_SOURCE レジスタの Activity が保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.INT_SOURCE` を指定した時に保持した値
または、SET 処理をした値

INT_SOURCE_DOUBLE_TAP プロパティ

概要

INT_SOURCE の DOUBLE_TAP を取得します。

処理内容

Read メソッドで `REG_ADDR.INT_SOURCE` を指定した時に INT_SOURCE レジスタの DOUBLE_TAP が保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.INT_SOURCE` を指定した時に保持した値
または、SET 処理をした値

INT_SOURCE_SINGLE_TAP プロパティ

概要

INT_SOURCE の SINGLE_TAP を取得します。

処理内容

Read メソッドで `REG_ADDR.INT_SOURCE` を指定した時に INT_SOURCE レジスタの SINGLE_TAP が保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.INT_SOURCE` を指定した時に保持した値
または、SET 処理をした値

INT_SOURCE_DATA_READY プロパティ

概要

INT_SOURCE の DATA_READY を取得します。

処理内容

Read メソッドで `REG_ADDR.INT_SOURCE` を指定した時に INT_SOURCE レジスタの DATA_READY が保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.INT_SOURCE` を指定した時に保持した値
または、SET 処理をした値

Range プロパティ

概要

Range を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.DATA_FORMAT` を指定した時に `DATA_FORMAT` レジスタの Range が保持されます。

Write メソッドで `REG_ADDR.DATA_FORMAT` を指定した時に Range 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.DATA_FORMAT` を指定した時に保持した値
または、SET 処理をした値

Justify プロパティ

概要

Justify を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.DATA_FORMAT` を指定した時に `DATA_FORMAT` レジスタの Justify が保持されます。

Write メソッドで `REG_ADDR.DATA_FORMAT` を指定した時に Justify 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.DATA_FORMAT` を指定した時に保持した値
または、SET 処理をした値

FULL_RES プロパティ

概要

FULL_RES を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.DATA_FORMAT` を指定した時に `DATA_FORMAT` レジスタの FULL_RES が保持されます。

Write メソッドで `REG_ADDR.DATA_FORMAT` を指定した時に FULL_RES 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.DATA_FORMAT` を指定した時に保持した値
または、SET 処理をした値

INT_INVERT プロパティ

概要

INT_INVERT を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.DATA_FORMAT` を指定した時に `DATA_FORMAT` レジスタの `INT_INVERT` が保持されます。

Write メソッドで `REG_ADDR.DATA_FORMAT` を指定した時に `INT_INVERT` 値を使用して書込みます。

SET 処理

`value` 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.DATA_FORMAT` を指定した時に保持した値
または、SET 処理をした値

SPI プロパティ

概要

SPI を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.DATA_FORMAT` を指定した時に `DATA_FORMAT` レジスタの SPI が保持されます。

Write メソッドで `REG_ADDR.DATA_FORMAT` を指定した時に SPI 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.DATA_FORMAT` を指定した時に保持した値
または、SET 処理をした値

SELF_TEST プロパティ

概要

SELF_TEST を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.DATA_FORMAT` を指定した時に `DATA_FORMAT` レジスタの `SELF_TEST` が保持されます。

Write メソッドで `REG_ADDR.DATA_FORMAT` を指定した時に `SELF_TEST` 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.DATA_FORMAT` を指定した時に保持した値
または、SET 処理をした値

DATAX プロパティ

概要

X-Axis Data を取得します。

処理内容

Read メソッドで `REG_ADDR.DATAX0` を指定した時に DATAX0 レジスタ、DATAX1 レジスタより X-Axis Data が計算され保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.DATAX0` を指定した時に保持した値
または、SET 処理をした値

DATAY プロパティ

概要

Y-Axis Data を取得します。

処理内容

Read メソッドで `REG_ADDR.DATAX0` を指定した時に `DATAY0` レジスタ、`DATAY1` レジスタより Y-Axis Data が計算され保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.DATAX0` を指定した時に保持した値
または、SET 処理をした値

DATAZ プロパティ

概要

Z-Axis Data を取得します。

処理内容

Read メソッドで `REG_ADDR.DATAX0` を指定した時に `DATAZ0` レジスタ、`DATAZ1` レジスタより Z-Axis Data が計算され保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.DATAX0` を指定した時に保持した値
または、SET 処理をした値

Samples プロパティ

概要

Samples を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.FIFO_CTL` を指定した時に `FIFO_CTL` レジスタの Samples が保持されます。

Write メソッドで `REG_ADDR.FIFO_CTL` を指定した時に Samples 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.FIFO_CTL` を指定した時に保持した値
または、SET 処理をした値

Trigger プロパティ

概要

Trigger を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.FIFO_CTL` を指定した時に `FIFO_CTL` レジスタの Trigger が保持されます。

Write メソッドで `REG_ADDR.FIFO_CTL` を指定した時に Trigger 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.FIFO_CTL` を指定した時に保持した値
または、SET 処理をした値

FIFO_MODE プロパティ

概要

FIFO_MODE を取得または設定します。

処理内容

Read メソッドで `REG_ADDR.FIFO_CTL` を指定した時に FIFO_CTL レジスタの FIFO_MODE が保持されます。

Write メソッドで `REG_ADDR.FIFO_CTL` を指定した時に FIFO_MODE 値を使用して書込みます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.FIFO_CTL` を指定した時に保持した値
または、SET 処理をした値

Entries プロパティ

概要

Entries を取得します。

処理内容

Read メソッドで `REG_ADDR.FIFO_STATUS` を指定した時に `FIFO_STATUS` レジスタの Entries が保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.FIFO_STATUS` を指定した時に保持した値
または、SET 処理をした値

FIFO_TRIG プロパティ

概要

FIFO_TRIG を取得します。

処理内容

Read メソッドで `REG_ADDR.FIFO_STATUS` を指定した時に `FIFO_STATUS` レジスタの `FIFO_TRIG` が保持されます。

SET 処理

value 値をセットする。

GET 処理

Read メソッドで `REG_ADDR.FIFO_STATUS` を指定した時に保持した値
または、SET 処理をした値

Init メソッド

概要

I2C バスコントローラに指定された接続設定を使って非同期的に接続します。

引数

| 引数名 | 型 | 説明 |
|---------|------|-----------|
| Address | byte | I2C のアドレス |

戻り値

なし

Read メソッド

概要

最新の変換結果を読み込みます。

引数

| 引数名 | 型 | 説明 |
|----------|------|--|
| REG_ADDR | byte | REG_ADDR_MODE1 (0x00) ~ REG_ADDR_ALLCALLADR (0x1B) |

戻り値

Status と同じ

1: 正常

1 以外: 異常

読込処理

| REG_ADDR 値 | 読込プロパティ |
|---------------|--|
| DEVID | DEVID |
| THRESH_TAP | THRESH_TAP |
| OFSX | OFSX OFSY OFSZ |
| DUR | DUR |
| Latent | Latent |
| Window | Window |
| THRESH_INACT | THRESH_INACT |
| TIME_INACT | TIME_INACT |
| ACT_INACT_CTL | INACT_Z_enable INACT_Y_enable INACT_X_enable INACT_acdc = ACT_Z_enable ACT_Y_enable ACT_X_enable ACT_acdc |

| REG_ADDR 値 | 読込プロパティ |
|----------------|---|
| THRESH_FF | THRESH_FF |
| TIME_FF | TIME_FF |
| TAP_AXES | TAP_Z_enable TAP_Y_enable TAP_X_enable Suppress |
| ACT_TAP_STATUS | TAP_Z_source TAP_Y_source TAP_X_source Asleep ACT_Z_source ACT_Y_source ACT_X_source |
| BW_RATE | Rate LOW_POWER |
| POWER_CTL | Wakeup Sleep Measure AUTO_SLEEP Link |
| INT_ENABLE | INT_ENABLE_Overrun INT_ENABLE_Watermark INT_ENABLE_FREE_FALL INT_ENABLE_Inactivity INT_ENABLE_Activity INT_ENABLE_DOUBLE_TAP INT_ENABLE_SINGLE_TAP INT_ENABLE_DATA_READY |

| REG_ADDR 値 | 読込プロパティ |
|-------------|---|
| INT_MAP | INT_MAP_Overrun INT_MAP_Watermark INT_MAP_FREE_FALL INT_MAP_Inactivity INT_MAP_Activity INT_MAP_DOUBLE_TAP INT_MAP_SINGLE_TAP INT_MAP_DATA_READY |
| INT_SOURCE | INT_SOURCE_Overrun INT_SOURCE_Watermark INT_SOURCE_FREE_FALL INT_SOURCE_Inactivity INT_SOURCE_Activity INT_SOURCE_DOUBLE_TAP INT_SOURCE_SINGLE_TAP INT_SOURCE_DATA_READY |
| DATA_FORMAT | Range Justify FULL_RES INT_INVERT SPI SELF_TEST |

| REG_ADDR 値 | 読込プロパティ |
|-------------|---------------------------------|
| DATA0 | DATA_X DATA_Y DATA_Z |
| FIFO_CTL | Samples Trigger FIFO_MODE |
| FIFO_STATUS | Entries FIFO_TRIG |

Write メソッド

概要

引数 REG_ADDR に関するプロパティを編集し書き込みます。

引数

| 引数名 | 型 | 説明 |
|----------|------|--|
| REG_ADDR | byte | REG_ADDR_MODE1 (0x00) ~ REG_ADDR_ALLCALLADR (0x1B) |

戻り値

Status

1: 正常

1 以外: 異常

書込処理

| REG_ADDR 値 | 書込プロパティ |
|---------------|--|
| DEVID | DEVID |
| THRESH_TAP | THRESH_TAP |
| OFSX | OFSX OFSY OFSZ |
| DUR | DUR |
| Latent | Latent |
| Window | Window |
| THRESH_INACT | THRESH_INACT |
| TIME_INACT | TIME_INACT |
| ACT_INACT_CTL | INACT_Z_enable INACT_Y_enable INACT_X_enable INACT_acdc = ACT_Z_enable ACT_Y_enable ACT_X_enable ACT_acdc |

| REG_ADDR 値 | 書込プロパティ |
|----------------|---|
| THRESH_FF | THRESH_FF |
| TIME_FF | TIME_FF |
| TAP_AXES | TAP_Z_enable TAP_Y_enable TAP_X_enable Suppress |
| ACT_TAP_STATUS | TAP_Z_source TAP_Y_source TAP_X_source Asleep ACT_Z_source ACT_Y_source ACT_X_source |
| BW_RATE | Rate LOW_POWER |
| POWER_CTL | Wakeup Sleep Measure AUTO_SLEEP Link |
| INT_ENABLE | INT_ENABLE_Overrun INT_ENABLE_Watermark INT_ENABLE_FREE_FALL INT_ENABLE_Inactivity INT_ENABLE_Activity INT_ENABLE_DOUBLE_TAP INT_ENABLE_SINGLE_TAP INT_ENABLE_DATA_READY |

| REG_ADDR 値 | 書込プロパティ |
|-------------|---|
| INT_MAP | INT_MAP_Overrun INT_MAP_Watermark INT_MAP_FREE_FALL INT_MAP_Inactivity INT_MAP_Activity INT_MAP_DOUBLE_TAP INT_MAP_SINGLE_TAP INT_MAP_DATA_READY |
| INT_SOURCE | INT_SOURCE_Overrun INT_SOURCE_Watermark INT_SOURCE_FREE_FALL INT_SOURCE_Inactivity INT_SOURCE_Activity INT_SOURCE_DOUBLE_TAP INT_SOURCE_SINGLE_TAP INT_SOURCE_DATA_READY |
| DATA_FORMAT | Range Justify FULL_RES INT_INVERT SPI SELF_TEST |

| REG_ADDR 値 | 書込プロパティ |
|-------------|---------------------------------|
| DATA0 | DATA_X DATA_Y DATA_Z |
| FIFO_CTL | Samples Trigger FIFO_MODE |
| FIFO_STATUS | Entries FIFO_TRIG |

サンプルプログラム

```
Imports I2CLibrary

Public NotInheritable Class ADXL345Page
    Inherits Page

    Private Const ADXL345_Address As Byte = &H53
    Private ADXL345 As ADXL345 = New ADXL345 ()
    Private WithEvents RefreshTimer As DispatcherTimer = New DispatcherTimer ()

Public Sub New()
    InitializeComponent()
    Call Init()
End Sub

Private Async Sub Init()
    Await ADXL345.Init(ADXL345_Address)
End Sub

Private Sub btnStart_Click(sender As Object, e As RoutedEventArgs)
    ADXL345.Read(ADXL345.REG_ADDR.DATA_FORMAT)
    ADXL345.Read(ADXL345.REG_ADDR.POWER_CTL)
    ADXL345.Read(ADXL345.REG_ADDR.BW_RATE)
    ADXL345.Read(ADXL345.REG_ADDR.INT_ENABLE)
    ADXL345.Read(ADXL345.REG_ADDR.TAP_AXES)

    ADXL345.Range = 0           ' +-2G
    ADXL345.Rate = 10          ' 100Hz
    ADXL345.Measure = 1       ' 測定モード
    ADXL345.INT_ENABLE_SINGLE_TAP = 1 ' シングルタップ検知
    ADXL345.TAP_X_enable = 1
    ADXL345.TAP_Y_enable = 1
    ADXL345.TAP_Z_enable = 1

```

```

ADXL345. Write (ADXL345. REG_ADDR. DATA_FORMAT)
ADXL345. Write (ADXL345. REG_ADDR. POWER_CTL)
ADXL345. Write (ADXL345. REG_ADDR. BW_RATE)
ADXL345. Write (ADXL345. REG_ADDR. INT_ENABLE)
ADXL345. Write (ADXL345. REG_ADDR. TAP_AXES)

```

```

ADXL345. THRESH_TAP = &H30          ' タップ割り込みのしきい値
ADXL345. DUR = &H10                ' 最大タップ持続時間

```

```

ADXL345. Write (ADXL345. REG_ADDR. THRESH_TAP)
ADXL345. Write (ADXL345. REG_ADDR. DUR)

```

```

RefreshTimer. Interval = TimeSpan.FromMilliseconds (100)
RefreshTimer. Start ()

```

End Sub

```

Private Sub Measurement (sender As Object, e As Object) Handles RefreshTimer.Tick
    RefreshTimer. Stop ()

```

```

ADXL345. Read (ADXL345. REG_ADDR. DATA0)
ADXL345. Read (ADXL345. REG_ADDR. INT_SOURCE)

```

```

Debug. WriteLine ("X-Axis : {0:D}", ADXL345. DATA0)
Debug. WriteLine ("Y-Axis : {0:D}", ADXL345. DATA1)
Debug. WriteLine ("Z-Axis : {0:D}", ADXL345. DATA2)

```

```

If ADXL345. INT_SOURCE_SINGLE_TAP = 1 Then
    Debug. WriteLine ("SINGLE_TAP detect")
End If

```

```

RefreshTimer. Start ()

```

End Sub

```

Private Sub btnStop_Click (sender As Object, e As RoutedEventArgs)
    ADXL345. Read (ADXL345. REG_ADDR. POWER_CTL)

```

```

ADXL345. Measure = 0

```

```
ADXL345. Write (ADXL345. REG_ADDR. POWER_CTL)
```

```
RefreshTimer. Stop ()
```

```
End Sub
```

```
Private Sub btnBack_Click(sender As Object, e As RoutedEventArgs)
```

```
    RefreshTimer. Stop ()
```

```
    ADXL345. close ()
```

```
    App. rootFrame. Navigate (GetType (MenuPage))
```

```
End Sub
```

```
End Class
```

4.最後に

このドキュメントは、I2C インターフェース・ドライバとして Windows 環境の VB で、I2C の各種デバイスを用いた開発を断念していた方々に、プロパティに値をセットしメソッドを呼ぶことで BIT 処理を実現し、VB で開発を可能としたドライバの解説書です。

本書は、I2C インターフェースドライバの解説書であり、各パラメータの説明書では有りません。ご使用する際は、該当する I2C デバイスのデータシートに記載されている

各レジスタと設定するパラメータと、その動作の関係を理解している事を前提に解説しています。

趣旨、ご理解の上ご利用頂ければ幸いです。

本製品の品質向上、対応する I2C デバイスの追加等のご案内を差し上げる為、ユーザー登録をお願いしております。

登録して頂いた情報につきましては、上記目的以外での利用は致しません。

ユーザー登録

| | |
|---------|-------------------------------------|
| 会社名 | |
| 住所 | |
| 所属部署 | |
| 役職 | |
| 氏名 | |
| E-mail | |
| 電話番号 | |
| FAX 番号 | |
| ご購入 I2C | |
| ご利用目的 | |
| ご案内 | 1.希望する 2.希望しない |
| ご意見 | |

FAX 又は E-MAIL にてお送り下さい。

960-0102 福島県福島市鎌田字川添 3-8 株式会社エフイーシー www.fec.co.jp

TEL 024-553-9611 FAX 024-553-9617

E-MAIL i2c@fec.co.jp