

ユーザーマニュアル

iR-COP

本マニュアルでは、iR-COP の仕様及び使用方法について説明します。

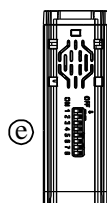
UM018001J_20200220

目次

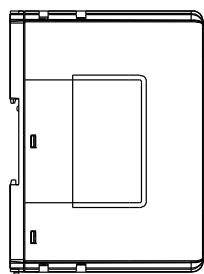
1. 製品外見	1
2. 製品仕様	2
3. LED インジケータ	3
3.1 L.V インジケータ	3
3.2 IO RUN/ERR インジケータ	3
3.3 CAN-RUN インジケータ	3
3.4 CAN-ERR インジケータ	3
4. 配置	3
4.1 ステーション番号設定	3
4.2 ボーレート設定 (DIP スイッチ)	4
4.3 CAN Bus 接続	4
4.4 CANOpen 機能	4
5. 通信エラーコントロール	4
5.1 紹介	4
5.2 Node Guarding Protocol	5
5.3 Heartbeat Protocol	6
5.4 エラー行為	7
6. オブジェクト辞書	9
6.1 Communication Profile Area 通信プロフィールエリア	9
6.1.1 【1000h】装置タイプ	10
6.1.2 【1001h】エラーレジスタ	10
6.1.3 【1002h】装置状態レジスタ	10
6.1.4 【1003h】エラーコード記録フィールド	11
6.1.5 【1005h】COB-ID メッセージを同期する	13
6.1.6 【1008h】メーカー装置名	13
6.1.7 【1009h】メーカーハードウェアバージョン	13
6.1.8 【100Ah】メーカーソフトウェアバージョン	13
6.1.9 【100Ch】Guard Time & 0x100D: Life Time Factor	13
6.1.10 【1010h】パラメータを保存する	13
6.1.11 【1011h】デフォルトパラメータに復元する	14
6.1.12 【1014h】COB-ID EMCY	22
6.1.13 【1015h】抑制時間 EMCY	22
6.1.14 【1016h】Consumer Heartbeat Time	22
6.1.15 【1017h】Producer Heartbeat Time	22
6.1.16 【1029h】エラー行為オブジェクト	22
6.2 PDO 通信パラメータ	23

6.2.1	RxPDO 通信パラメータ	23
6.2.2	TxPDO 通信パラメータ	24
6.2.3	PDO マッピングパラメータ	26
6.2.4	マッピングパラメータ	33
6.3	I/O 装置オブジェクト辞書エリア(DS401).....	33
6.3.1	デジタル入力点を読み取る	33
6.3.2	デジタル出力点を書き込む.....	34
6.3.3	アナログ入力チャンネルを読み取る	35
6.3.4	アナログ出力チャンネルを書き込む (64 channel)	35
6.4	メーカー特定プロフィールエリア	36
6.4.1	2000h-2001h : デジタル入力フィルタ	36
6.4.2	3000h~300Fh: モジュールレジスタ	36
6.4.3	4000h-4007h : iBus 及びモジュール情報	37
7.	消費電力	38
8.	CODESYS で Remote I/O に接続する	39
9.	CODESYS でアナログモジュールを設定する	39
9.1	SDO を追加する	39
9.1.1	アナログ入力パラメータを書き込む.....	39
9.1.2	アナログ出力パラメータを書き込む.....	40
9.1.3	温度入力パラメータを書き込む.....	41
9.2	Weintek_Library 機能ブロック	42
9.2.1	Analog_Config – 1 個のレジスタを読み取り/書き込み.....	42
9.2.2	AI_Ch_Pa – アナログ入力のチャンネルをに読み取る・書き込む .44	
9.2.3	AO_Ch_Pa – アナログ出力のチャンネルに読み取る・書き込む....	46
9.2.4	Analog_VI_Read – アナログモジュールの全てのレジスタを読み取る	
	48	

1. 製品外見

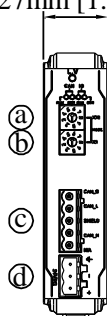


上面図

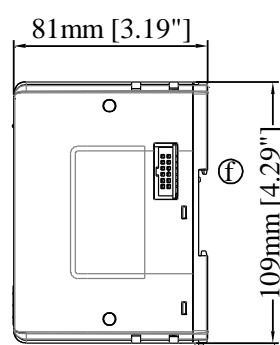


側面図

27mm [1.06"]



正面図



側面図



下面図

<i>a</i>	ステーション番号ロータ リースイッチ x10	<i>e</i>	ボーレート DIP スイッチ
<i>b</i>	ステーション番号ロータ リースイッチ x1	<i>f</i>	拡張コネクタ
<i>c</i>	CAN bus コネクタ		
<i>d</i>	電源コネクタ		

2. 製品仕様

通信インターフェース仕様							
拡張 I/O モジュール	接続数		内部バスの電流による				
	デジタル入力点数		Max. 256				
	デジタル出力点数		Max. 128				
	アナログ入力チャンネル数		Max. 64				
	アナログ出力チャンネル数		Max. 64				
インジケータ	CAN RUN (緑)		CANopen 状態インジケータ				
	CAN ERR (赤)		CANopen エラーインジケータ				
	L.V (赤)		低電圧インジケータ				
	IO RUN (緑)		モジュール状態インジケータ				
	IO ERR (赤)		モジュールエラーインジケータ				
データ送信速度 (bps)	1M	800k	500k	250k	125k	100k	50k
ケーブル長	20m	50m	100m	250m	500m	600m	1,000m
PDOs 数(CANopen)	8 TxPDOs / 8 RxPDOs						
データ処理の操作モード	同期、イベント駆動、イベントタイマー、ポーリング						
SDOs 数	1 Standard SDOs						
バス接続	オープンスタイル 5 ピンプラグ x 1						
CANopen 付加機能	life/node guarding, heartbeat, emergency object, variables mapping, store/restore, output error mode.						
一般的な仕様							
電源	入力電圧		24 VDC (-15%/+20%)				
	消費電流		名目上 24VDC@ 100mA				
	内部バス電流		Max 2A @ 5VDC				
	COP の消費電流		170mA @ 5VDC				
	電源絶縁		CanOpen 通信部: 有り 電源部: 有り				
	バックアップフューズ		≤1.6A リセットブルヒューズ				
仕様	PCB コーティング		有り				
	筐体材質		プラスチック				
	外形寸法 WxHxD		27 x 109 x 81 mm				
	重量		約 0.15 kg				
	取付		35mm DIN レール取付				
使用環境	保護等級		IP20				
	保存温度		-20° ~ 70°C (-4° ~ 158°F)				
	使用温度		0° ~ 55°C (32° ~ 131°F)				
	使用湿度		10% ~ 90% (結露無き事)				
接続ケーブル	線の太さ		0.5 mm ² ... 2.5 mm ² 、標準、単線、AWG 26-12				
認証	EMC Immunity		EN 55032: 2012+AC: 2013, Class A EN 61000-6-4: 2007+A1:2011 EN 55024: 2010+A1: 2015 EN 61000-6-2:2005 に準拠				

3. LED インジケータ

3.1 L.V インジケータ

L.V LED 状態	記述
OFF	24VDC電源が正常
点滅	24VDC電源を確認中
ON	24VDC電源が異常

3.2 IO RUN/ERR インジケータ

RUN LED	ERR LED	記述
OFF	OFF	電源が切れた
点滅	OFF	I/Oが初期化中
点滅	ON	I/Oの初期化に失敗した
ON	OFF	I/Oが正常稼働中
ON	点滅	I/Oモジュールアラーム
ON	ON	I/O通信に失敗した

3.3 CAN-RUN インジケータ

NO	CAN-RUN LED	状態	記述
1	点灯	操作	装置は操作状態にいる
2	点滅	前操作	装置は前操作状態にいる
3	1 回点滅	停止	装置は停止状態にいる

3.4 CAN-ERR インジケータ

NO	CAN-ERR LED	状態	記述
1	ON	CAN bus off	CAN bus は現在 off 状態にいる
2	3 回点滅	SYNC エラー	設定された通信期間内、SYNC のメッセージを受信していなかった(オブジェクトディクショナリ エントリ 0x1006 をご参照)
3	2 回点滅	イベント発生エラー	guarding イベントまたは heartbeat イベントが発生した
4	1 回点滅	エラー数ウォーニング	CAN bus エラー数がウォーニングに達した
5	点滅	設定エラー	一般的な設定が正しくなかった
6	OFF	エラー無し	エラー無し。稼働状態にいる

4. 配置

4.1 ステーション番号設定

ロータリースイッチでステーション番号を設定できます。範囲は 1 ~ 99 (0 をサポートしません)です。



設定	記述
0	無効なノードID
1~99	有効なノードID

4.2 ボーレート設定 (DIP スイッチ)

SW4	SW3	SW2	SW1	ボーレート
0	0	0	0	ボーレート 自動検出
0	0	0	1	1Mbps
0	0	1	0	800Kbps
0	0	1	1	500Kbps
0	1	0	0	250Kbps
0	1	0	1	125Kbps
0	1	1	0	100Kbps
0	1	1	1	50Kbps
SW5-7	予約済み			
SW8	CAN Bus 120Ω 終端抵抗			

4.3 CAN Bus 接続



PIN#	名前
1	CAN_G
2	CAN_L
3	SHIELD
4	CAN_H
5	N/A

4.4 CANOpen 機能

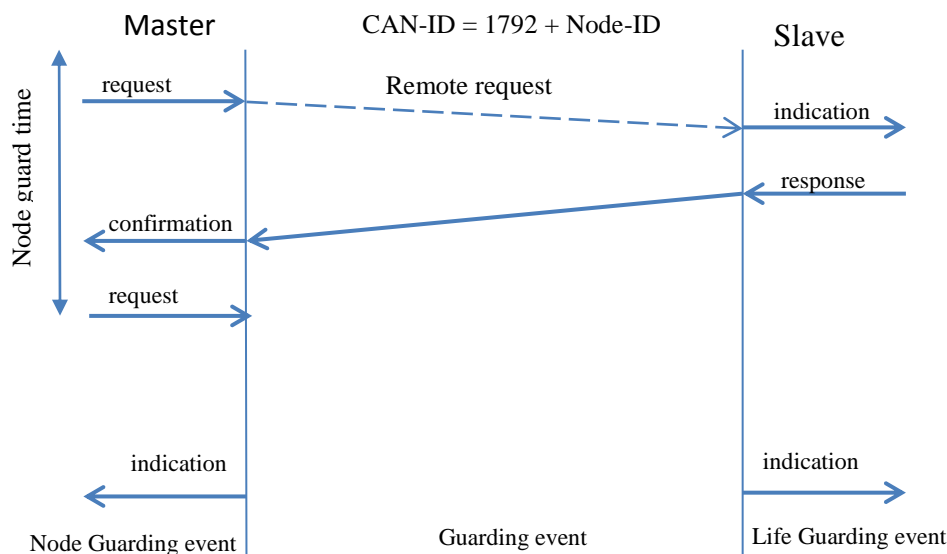
- 8 TxPDO s
- 8 RxPDO
- 1 standard SDO
- 1 emergency object (EMCY)
- 1 synchronization object (SYNC, without time stamp)
- Guarding
- Heartbeat
- NMT objects

5. 通信エラーコントロール

5.1 紹介

CANopen エラーコントロールプロトコル(Error Control Protocols)は 2 種類があり、Guarding Protocol 及び Heartbeat Protocol です。しかし、同時に使用することができないので、同一時間ではこの二つのプロトコルから一つを選んで使用してください。

5.2 Node Guarding Protocol



一定時間(Node Guard Time)ごとに、NMT マスターは Remote transmit request を送信し、各 NMT マスターにポーリング(POLL)を行います。ユーザーは各 NMT スレーブに対して異なる Node Guard Time を設定することができます。NMT スレーブが Remote Transmit request を受信すると、現在の状態:前操作、操作、または停止を応答します。応答されたデータの形式は以下の通りです：

Bit7	Bit6~Bit0
Toggle bit.	4 : 停止 5 : 操作 127 : 前操作

Node Life Time は 2 個のパラメータ：100Ch (Guard time)、100Dh (life time factor)で構成されます。

$$\text{Node Life Time} = \text{Life Time Factor} \times \text{Guard Time (ms)}$$

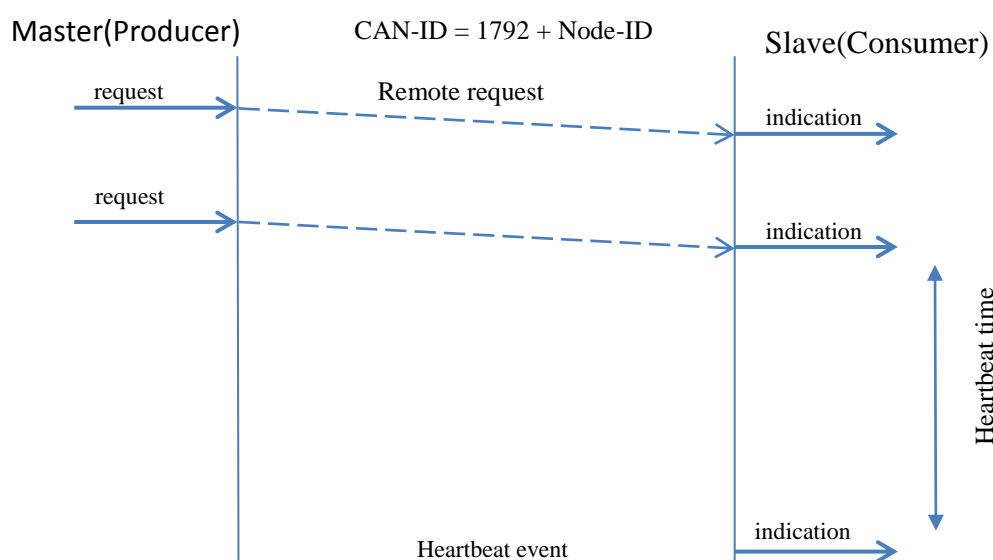
Guard Time				
インデックス	サブインデックス	データ型	デフォルト値	記述
100Ch	00h	UNSIGNED16	0000h	0 : 無効にする

Life Time Factor				
インデックス	サブインデックス	データ型	デフォルト値	記述
100Dh	00h	UNSIGNED8	00h	0 : 無効にする

Node Life Time 時間内で NMT マスターからのリクエストを受信していない場合、NMT スレーブは Life Guarding Event を生成します。

5.3 Heartbeat Protocol

Heartbeat Protocol はエラーコントロールサービスを同一のメッセージに包み込んで發送します。Heartbeat Producer は一定時間(Heartbeat Consumer Time)で Heartbeat メッセージを Heartbeat Consumer に送信します。Heartbeat Consumer は一定時間(Heartbeat Consumer Time)で Heartbeat メッセージを消費します。Heartbeat Consumer Time 時間内でメッセージを受信していない場合、Heartbeat Consumer は Heartbeat Event を生成します。



Producer Heartbeat Time				
インデックス	サブインデックス	データ型	デフォルト値	記述
1017h	00h	UNSIGNED16	00h	0 : 無効にする (単位 : ms)

Consumer Heartbeat Time				
インデックス	サブインデックス	データ型	デフォルト値	記述
1016h	00h	UNSIGNED32	00h	---

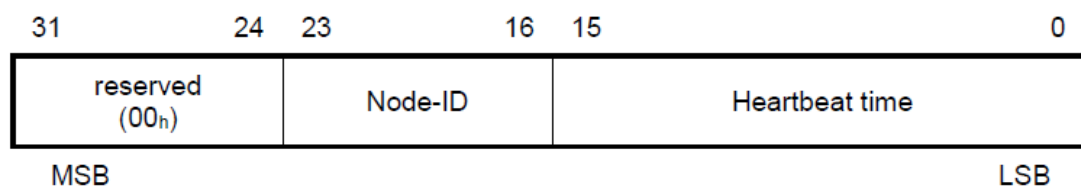


Figure 62: Structure of Consumer heartbeat time

- ノード ID: ID が 0 または 127 より大きい場合、heartbeat が無効されます。
- Heartbeat time の時間単位: ms

注意: consumer heartbeat time を producer heartbeat time より大きく設定する必要があります。

5.4 エラー行為

Life Guarding イベントまたは Heartbeat イベントが発生した場合:

状態は直接にオブジェクトディクショナリエントリを通じて設定されます(Error Behavior Object)。

Error Behavior Object		
インデックス	サブインデックス	記述
1029h	01h	0 : NMT 状態を前操作に変更する(現在、NMT は操作可能状態である場合) 1 : NMT 状態を変更しない 2 : NMT を停止状態に変更する

エラー状態下、デジタル及びアナログ出力はエラー出力モード及びエラー出力値で配置され、出力状態を定義できます。

もしイベントが発生する際にエラー出力モードが有効されると、出力値はエラー値で与えられます。

イベントが発生する際にエラー出力モードが無効されると、出力値は維持されます。

インデックス 6206 : エラー出力モード-デジタル出力 8-bit			
サブインデックス	記述		デフォルト
00h	サポートするサブインデックス最大数		
01h	エラー出力モード-デジタル出力点 01h to 08h	0 : 出力値を維持する	FFh
02h	エラー出力モード-デジタル出力点 09h to 10h	1 : 0D6207 の値で出力する	FFh

03-FEh	エラー出力モード-デジタル出力点 11h to 7F0h		FFh
--------	------------------------------	--	-----

インデックス 6207 : エラー出力値-デジタル出力値 8-bit			
サブインデックス	記述		デフォルト
00h	サポートするサブインデックス最大数		
01h	エラー出力値-デジタル出力値 01h to 08h		00h
02h	エラー出力値-デジタル出力値 09h to 10h		00h
03h-FEh	エラー出力値-デジタル出力値 11h to 7F0h		00h

インデックス 6443 : エラー出力モード-アナログ出力			
サブインデックス	記述		デフォルト
00h	サポートするサブインデックス最大数		
01h	エラー出力モード-アナログ出力 01h	0 : 出力値を保持する 1: 6444h の値で出力する その他 = 予約済み	01h
02h	エラー出力モード-アナログ出力 02h		01h
03h-FEh	エラー出力モード-アナログ出力 03h-FEh		01h

インデックス 6444 : エラー出力値-アナログ出力値		
サブインデックス	記述	デフォルト
00h	サポートするサブインデックス最大数	
01h	エラー出力値-アナログ出力値 01h	0000 000h
02h	エラー出力値-アナログ出力値 01h	0000 000h
03h-FEh	エラー出力値-アナログ出力値 03h-FEh	0000 000h

6. オブジェクト辞書

Object Area	インデックス範囲 (hex)
通信プロフィールエリア	1000-1FFF
メーカープロフィールエリア	2000-5FFF
標準装置プロフィールエリア	6000-9FFF

6.1 Communication Profile Area 通信プロフィールエリア

インデックス	サブインデックス	記述	タイプ	属性	デフォルト
1000h	00h	装置タイプ	UNSIGNED32	ro	---
1001h	00h	エラーレジスタ	UNSIGNED8	ro	0
1002h	00h	メーカー状態レジスタ	UNSIGNED32	ro	0
	01h	モジュールアラーム	UNSIGNED32	ro	0
	02h	モジュールが切断された	UNSIGNED32	ro	0
	03h	iBus 初期化エラーコード	UNSIGNED32	ro	0
1003h	00h	エラーコードサブインデックス数	UNSIGNED32	rw	0
	01h	緊急エラーコード (最新)	UNSIGNED32	ro	0
	02h-3Fh	緊急エラーコード	UNSIGNED32	ro	0
	40h	緊急エラーコード (最新)	UNSIGNED32	ro	0
1005h	00h	COB-ID メッセージを同期する	UNSIGNED32	ro	00000080h
1008h	00h	メーカー装置名	STRING	ro	'iR-COP'
1009h	00h	メーカーハードウェアバージョン	STRING	ro	'1.00.0'
100Ah	00h	メーカーソフトウェアバージョン	STRING	ro	'1.00.0'
100Ch	00h	Guard time	UNSIGNED16	rw	0
100Dh	00h	Life time factor	UNSIGNED8	rw	0
1014h	00h	COB-ID EMCY	UNSIGNED32	rw	80h + Node-ID
1015h	00h	抑制時間 EMCY	UNSIGNED16	rw	0
1016h	00h	サブインデックス数	UNSIGNED8	ro	0
	01h	Consumer heartbeat time	UNSIGNED32	rw	0
1017h	00h	Producer heartbeat time	UNSIGNED16	rw	0
1018h		製品識別オブジェクト			
	00h	サブインデックス数	UNSIGNED8	ro	04h
	01h	メーカーID	UNSIGNED32	ro	0000044Eh
	02h	製品コード	UNSIGNED32	ro	00000701h
	03h	バージョン番号	UNSIGNED32	ro	---

	04h	シリアル番号	UNSIGNED32	ro	---
1027h	00h	接続モジュール数	UNSIGNED8	ro	01h – 10h
	01h-10h	装置コードモジュール 1-16	UNSIGNED16	ro	----
1029h		エラー行為オブジェクト			
	00h	サブインデックスの数	UNSIGNED8	ro	01h
	01h	通信エラー	UNSIGNED8	rw	0

6.1.1 【1000h】装置タイプ

MSB			LSB		
31	24	23	16	15	0
付加情報				一般情報	
Specific functionality		I/O functionality		Device profile number	

Specific functionality: iR-COP モジュールは特定機能を持っていないため、0 に維持します。

General information: ドキュメント DS-401 に準拠し、191h です。

I/O functionality:

bit 16 は 1 の場合: デジタル入力チャンネルが存在します。

bit 17 は 1 の場合: デジタル出力チャンネルが存在します。

bit 18 は 1 の場合: アナログ入力チャンネルが存在します。

bit 19 は 1 の場合: アナログ出力チャンネルが存在します。

Bit 20 ~ 23 確保: 数値は 1 です。

6.1.2 【1001h】エラーレジスタ

エラーレジスタです。詳細は 1003h にご参考ください。

Bit	記述
0	一般的なエラー
1	電流エラー
2	電圧エラー
3	温度エラー
4	通信エラー
5	装置プロフィールに定義されたエラー
6	予約済み(0)
7	メーカー定義

6.1.3 【1002h】装置状態レジスタ

Bit	記述
0	低電力アラーム
1	ハードウェアエラー

2	予約済み
3	Heartbeatイベント
4	Guardingイベント
5	CAN エラーパッシブモード
6	CANオーバーラン(overrun)
7	モジュールが通信を失った
8	モジュールアラーム
9	iBusの初期化に失敗した
10	iBusの数が16を超えた
11	iBusシステムでの消費電力が超えた

- サブインデックス 01: モジュールアラーム
Bit0 は 1 の場合、一番目のモジュールがアラームを出したと示し、Bit2 は 1 の場合、二番目のモジュールモジュールがアラームを出したと示し、このように類推します。
- サブインデックス 02: モジュールが接続を失った
- Bit0 は 1 の場合、一番目のモジュールが通信を失ったと示し、Bit2 は 1 の場合、二番目のモジュールが通信を失ったと示し、このように類推します。サブインデックス 03: iBus iBus 初期化エラー
- サブインデックス 01: エラーコード(Error bit)

6.1.4 【1003h】エラーコード記録フィールド

エラーが発生した時、Emergency Object が生成され、エラーコード記録フィールドに記録され、エラー履歴を提供します。

サブインデックス 0 はエラー記録の数で、サブインデックス 1 からエラー記録です。サブインデックス 0 に 0 を書き込むと、エラー記録が消去されます。0 しか書き込めなく、0 以外の数値を書き込むと、SDO は abort 0609 0030h と応答します。

エラー記録コードフォーマット: bit0~15 はエラーコードで、bit16~31 は付加情報です。

Byte:

MSB

LSB

付加情報	エラーコード
------	--------

Emergency Error Codes:

エラーコード (hex)	記述
00xx	エラーリセット/エラー無し
10xx	一般的なエラー
20xx	電流
21xx	電流、装置入力側
22xx	電流、装置内部
23xx	電流、装置出力側
30xx	電圧
31xx	主電圧
32xx	装置内部電圧
33xx	出力電圧
40xx	温度
41xx	周囲温度
42xx	装置温度
50xx	装置ハードウェア
60xx	装置ソフトウェア
61xx	内部ソフトウェア
62xx	ユーザーソフトウェア
63xx	データ設定
70xx	追加モジュール
80xx	監視
81xx	通信エラー
8110	CAN オーバーラン(オブジェクト紛失)
8120	CAN エラーパッシブモード
8130	Life Guardingエラーまたは Heartbeatエラー
8140	bus offから復元した
8150	COB-ID送信コリジョン
82xx	プロトコルエラー
8210	PDOデータ長さエラー
8220	PDOデータ長さが長すぎた
90xx	外部エラー
F0xx	追加機能エラー
FFxx	装置仕様エラー

iR-COP エラーレジスタ

エラーレジスタ	Predefined Error Field	記述
01h	3100h	低電力アラーム
01h	5000h	ハードウェアエラー
10h	8100h	CAN Bus off (予約済み)
10h	8130h	Heartbeatイベント
10h	8130h	Guardingイベント
10h	8120h	CANエラーパッシブモード
10h	8110h	CANオーバーラン
80h	7000h	モジュールが通信を失った
80h	7001h	モジュールアラーム
80h	7002h	iBusの初期化に失敗した
80h	7003h	ibusの数が16を超えた
80h	7004h	ibusシステムでの消費電力を超えた

6.1.5 【1005h】 COB-ID メッセージを同期する

COB-ID はメッセージの同期に用いられます。

Bit0~10: COB-ID を同期する

Bit11~31: iR-COP は 0

6.1.6 【1008h】 メーカー装置名

ストリングで、装置の名前‘iR-COP’を表示します。

6.1.7 【1009h】 メーカーハードウェアバージョン

ストリングで、装置のハードウェアバージョン‘1.00.0’を表示します。

6.1.8 【100Ah】 メーカーソフトウェアバージョン

ストリングで、装置のソフトウェアバージョン‘1.00.0’を表示します。

6.1.9 【100Ch】 Guard Time & 0x100D: Life Time Factor

Guard time 及び life time factor は life guarding protocol に用いられます。0 に設定すると、guarding 機能が無効されます。

Life Time = Life Time Factor * Guard Time (ms)

(関連の説明については、Node Guarding Protocol にご参考ください)

6.1.10 【1010h】 パラメータを保存する

本オブジェクトは不揮発性メモリー内のパラメータの保存を制御することに用いられます。

VALUE DEFINITION

- サブインデックス 01h: 全ての設定可能なパラメータを保存します。
- サブインデックス 02h: 通信関連のパラメータ(index from 1000_h to 1FFF_h)を保存します。

- サブインデックス 03h: アプリケーション関連のパラメータ(index from 6000_h to 9FFF_h)を保存します。

MSB		LSB	
e	v	a	s
65 _h	76 _h	61 _h	73 _h

Storage write access signature

6.1.11 【1011h】デフォルトパラメータに復元する

このオブジェクトを使用すると、通信プロファイル、装置プロファイル、およびアプリケーションプロファイルに対応するパラメータがデフォルト値に復元されます。

サブインデックス 01h: 復元可能な全てのパラメータです。

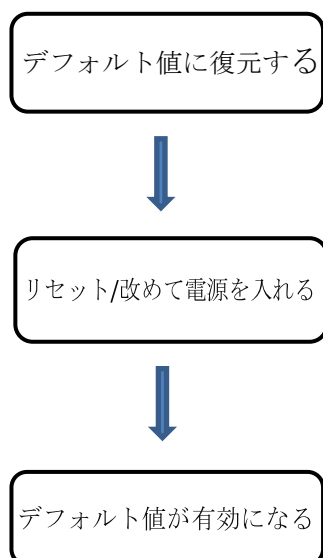
サブインデックス 02h: 通信関連のパラメータ(index from 1000_h to 1FFF_h)です。

サブインデックス 03h: アプリケーション関連のパラメータ(index from 6000_h to 9FFF_h)です。

MSB		LSB	
d	a	o	l
64 _h	61 _h	6F _h	6C _h

Restore default write access signature

デフォルト値をCANopen装置がリセットされた後(01hから7FhのサブインデックスのNMTサービスリセットノード、サブインデックス02hのNMTサービスリセット通信)、または電源が入れ直された後で有効に設定する必要があります。



通信パラメータ(1000h~1FFFh):

インデックス	サブインデックス	記述	タイプ	デフォルト
100Ch	00h	Guard time	UNSIGNED16	0
100Dh	00h	Life time factor	UNSIGNED8	0
1014h	00h	EMCY COB-ID	UNSIGNED32	80h + Node-ID
1015h	00h	EMCY 抑制時間	UNSIGNED16	0
1016h	01h	Consumer heartbeat time	UNSIGNED32	0
1017h	00h	Producer heartbeat time	UNSIGNED16	0
1029h	01h	通信エラー	UNSIGNED8	0
1400h	01h	RxPDO1 の COB-ID	UNSIGNED32	200h + Node-ID
	02h	RxPDO1 の送信タイプ	UNSIGNED8	FFh
1401h	01h	RxPDO2 の COB-ID	UNSIGNED32	300h + Node-ID
	02h	RxPDO2 の送信タイプ	UNSIGNED8	FFh
1402h	01h	RxPDO3 の COB-ID	UNSIGNED32	400h + Node-ID
	02h	RxPDO3 の送信タイプ	UNSIGNED8	FFh
1403h	01h	RxPDO4 の COB-ID	UNSIGNED32	500h + Node-ID
	02h	RxPDO4 の送信タイプ	UNSIGNED8	FFh
1404h-1407h	01h	RxPDO5-8 の COB-ID	UNSIGNED32	8000000
	02h	RxPDO5-8 の送信タイプ	UNSIGNED8	FFh
1800h	01h	TxPDO1 の COB-ID	UNSIGNED32	180h + Node-ID
	02h	TxPDO1 の送信タイプ	UNSIGNED8	FFh
	03h	TxPDO1 の禁制時間	UNSIGNED16	0
	05h	TxPDO1 のイベントタイマー	UNSIGNED16	0
1801h	01h	TxPDO2 の COB-ID	UNSIGNED32	280h + Node-ID
	02h	TxPDO2 の送信タイプ	UNSIGNED8	FFh
	03h	TxPDO2 の禁制時間	UNSIGNED16	0
	05h	TxPDO2 のイベントタイマー	UNSIGNED16	0
1802h	01h	TxPDO3 の COB-ID	UNSIGNED32	380h + Node-ID
	02h	TxPDO3 の送信タイプ	UNSIGNED8	FFh
	03h	TxPDO3 の禁制時間	UNSIGNED16	0
	05h	TxPDO3 のイベントタイマー	UNSIGNED16	0
1803h	01h	TxPDO4 の COB-ID	UNSIGNED32	480h + Node-ID
	02h	TxPDO4 の送信タイプ	UNSIGNED8	FFh
	03h	TxPDO4 の禁制時間	UNSIGNED16	0

	05h	TxPDO4 のイベントタイマー	UNSIGNED16	0
1804h-7h	01h	TxPDO5-8 の COB-ID	UNSIGNED32	280h + Node-ID
	02h	TxPDO5-8 の送信タイプ	UNSIGNED8	FFh
	03h	TxPDO5-8 の禁制時間	UNSIGNED16	0
	05h	TxPDO5-8 のイベントタイマー	UNSIGNED16	0
1600h	00h	RPDO1 マッピングパラメータ、サポートするサブインデックス最大数	UNSIGNED8	08h
	01h	第 1 セットのアプリケーションオブジェクトセットのアプリケーションオブジェクト	UNSIGNED32	6200 01 08h
	02h	第 2 セットのアプリケーションオブジェクト	UNSIGNED32	6200 02 08h
	03h	第 3 セットのアプリケーションオブジェクト	UNSIGNED32	6200 03 08h
	04h	第 4 セットのアプリケーションオブジェクト	UNSIGNED32	6200 04 08h
	05h	第 5 セットのアプリケーションオブジェクト	UNSIGNED32	6200 05 08h
	06h	第 6 セットのアプリケーションオブジェクト	UNSIGNED32	6200 06 08h
	07h	第 7 セットのアプリケーションオブジェクト	UNSIGNED32	6200 07 08h
	08h	第 8 セットのアプリケーションオブジェクト	UNSIGNED32	6200 08 08h
1601h	00h	RPDO2 マッピングパラメータ、サポートするサブインデックス最大数	UNSIGNED8	04h
	01h	第 1 セットのアプリケーションオブジェクト	UNSIGNED32	6411 01 10h
	02h	第 2 セットのアプリケーションオブジェクト	UNSIGNED32	6411 02 10h
	03h	第 3 セットのアプリケーションオブジェクト	UNSIGNED32	6411 03 10h
	04h	第 4 セットのアプリケーションオブジェクト	UNSIGNED32	6411 04 10h
1602h	00h	RPDO3 マッピングパラメータ、	UNSIGNED8	04h

		サポートするサブインデックス 最大数		
	01h	第 1 セットのアプリケーション オブジェクト	UNSIGNED32	6411 05 10h
	02h	第 2 セットのアプリケーション オブジェクト	UNSIGNED32	6411 06 10h
	03h	第 3 セットのアプリケーション オブジェクト	UNSIGNED32	6411 07 10h
	04h	第 4 セットのアプリケーション オブジェクト	UNSIGNED32	6411 08 10h
1603h	00h	RPDO4 マッピングパラメータ、 サポートするサブインデックス 最大数	UNSIGNED8	04h
	01h	第 1 セットのアプリケーション オブジェクト	UNSIGNED32	6411 09 10h
	02h	第 2 セットのアプリケーション オブジェクト	UNSIGNED32	6411 0A 10h
	03h	第 3 セットのアプリケーション オブジェクト	UNSIGNED32	6411 0B 10h
	04h	第 4 セットのアプリケーション オブジェクト	UNSIGNED32	6411 0C 10h
1604h	00h	RPDO5 マッピングパラメータ、 サポートするサブインデックス 最大数	UNSIGNED8	08h
	01h	第 1 セットのアプリケーション オブジェクト	UNSIGNED32	6200 09 08h
	02h	第 2 セットのアプリケーション オブジェクト	UNSIGNED32	6200 0A 08h
	03h	第 3 セットのアプリケーション オブジェクト	UNSIGNED32	6200 0B 08h
	04h	第 4 セットのアプリケーション オブジェクト	UNSIGNED32	6200 0C 08h
	05h	第 5 セットのアプリケーション オブジェクト	UNSIGNED32	6200 0D 08h
	06h	第 6 セットのアプリケーション オブジェクト	UNSIGNED32	6200 0E 08h
	07h	第 7 セットのアプリケーション	UNSIGNED32	6200 0F 08h

		オブジェクト		
	08h	第 8 セットのアプリケーション オブジェクト	UNSIGNED32	6200 10 08h
1605h	00h	RPDO6 マッピングパラメータ、 サポートするサブインデックス 最大数	UNSIGNED8	04h
	01h	第 1 セットのアプリケーション オブジェクト	UNSIGNED32	6411 0D 10h
	02h	第 2 セットのアプリケーション オブジェクト	UNSIGNED32	6411 0E 10h
	03h	第 3 セットのアプリケーション オブジェクト	UNSIGNED32	6411 0F 10h
	04h	第 4 セットのアプリケーション オブジェクト	UNSIGNED32	6411 10 10h
1606h	00h	RPDO7 マッピングパラメータ、 サポートするサブインデックス 最大数	UNSIGNED8	04h
	01h	第 1 セットのアプリケーション オブジェクト	UNSIGNED32	6411 11 10h
	02h	第 2 セットのアプリケーション オブジェクト	UNSIGNED32	6411 12 10h
	03h	第 3 セットのアプリケーション オブジェクト	UNSIGNED32	6411 13 10h
	04h	第 4 セットのアプリケーション オブジェクト	UNSIGNED32	6411 14 10h
1607h	00h	RPDO8 マッピングパラメータ、 サポートするサブインデックス 最大数	UNSIGNED8	04h
	01h	第 1 セットのアプリケーション オブジェクト	UNSIGNED32	6411 15 10h
	02h	第 2 セットのアプリケーション オブジェクト	UNSIGNED32	6411 16 10h
	03h	第 3 セットのアプリケーション オブジェクト	UNSIGNED32	6411 17 10h
	04h	第 4 セットのアプリケーション オブジェクト	UNSIGNED32	6411 18 10h
1A00h	00h	TPDO1 マッピングパラメータ、	UNSIGNED8	08h

		サポートするサブインデックス 最大数		
	01h	第 1 セットのアプリケーション オブジェクト	UNSIGNED32	6000 01 08h
	02h	第 2 セットのアプリケーション オブジェクト	UNSIGNED32	6000 02 08h
	03h	3 セットのアプリケーションオ ブジェクト	UNSIGNED32	6000 03 08h
	04h	第 4 セットのアプリケーション オブジェクト	UNSIGNED32	6000 04 08h
	05h	第 5 セットのアプリケーション オブジェクト	UNSIGNED32	6000 05 08h
	06h	第 6 セットのアプリケーション オブジェクト	UNSIGNED32	6000 06 08h
	07h	第 7 セットのアプリケーション オブジェクト	UNSIGNED32	6000 07 08h
	08h	第 8 セットのアプリケーション オブジェクト	UNSIGNED32	6000 08 08h
1A01h	00h	TPDO2 マッピングパラメータ、 サポートするサブインデックス 最大数	UNSIGNED8	04h
	01h	第 1 セットのアプリケーション オブジェクト	UNSIGNED32	6401 01 10h
	02h	第 2 セットのアプリケーション オブジェクト	UNSIGNED32	6401 02 10h
	03h	第 3 セットのアプリケーション オブジェクト	UNSIGNED32	6401 03 10h
	04h	第 4 セットのアプリケーション オブジェクト	UNSIGNED32	6401 04 10h
1A02h	00h	TPDO3 マッピングパラメータ、 サポートするサブインデックス 最大数	UNSIGNED8	04h
	01h	第 1 セットのアプリケーション オブジェクト	UNSIGNED32	6401 05 10h
	02h	第 2 セットのアプリケーション オブジェクト	UNSIGNED32	6401 06 10h
	03h	第 3 セットのアプリケーション	UNSIGNED32	6401 07 10h

		オブジェクト		
	04h	第 4 セットのアプリケーション オブジェクト	UNSIGNED32	6401 08 10h
1A03h	00h	TPDO4 マッピングパラメータ、 サポートするサブインデックス 最大数	UNSIGNED8	04h
	01h	第 1 セットのアプリケーション オブジェクト	UNSIGNED32	6401 09 10h
	02h	第 2 セットのアプリケーション オブジェクト	UNSIGNED32	6401 0A 10h
	03h	第 3 セットのアプリケーション オブジェクト	UNSIGNED32	6401 0B 10h
	04h	第 4 セットのアプリケーション オブジェクト	UNSIGNED32	6401 0C 10h
1A04h	00h	TPDO5 マッピングパラメータ、 サポートするサブインデックス 最大数	UNSIGNED8	08h
	01h	第 1 セットのアプリケーション オブジェクト	UNSIGNED32	6000 09 08h
	02h	第 2 セットのアプリケーション オブジェクト	UNSIGNED32	6000 0A 08h
	03h	第 3 セットのアプリケーション オブジェクト	UNSIGNED32	6000 0B 08h
	04h	第 4 セットのアプリケーション オブジェクト	UNSIGNED32	6000 0C 08h
	05h	第 5 セットのアプリケーション オブジェクト	UNSIGNED32	6000 0D 08h
	06h	第 6 セットのアプリケーション オブジェクト	UNSIGNED32	6000 0E 08h
	07h	第 7 セットのアプリケーション オブジェクト	UNSIGNED32	6000 0F 08h
	08h	第 8 セットのアプリケーション オブジェクト	UNSIGNED32	6000 10 08h
1A05h	00h	TPDO6 マッピングパラメータ、 サポートするサブインデックス 最大数	UNSIGNED8	04h
	01h	第 1 セットのアプリケーション	UNSIGNED32	6401 0D 10h

		オブジェクト		
	02h	第 2 セットのアプリケーション オブジェクト	UNSIGNED32	6401 0E 10h
	03h	第 3 セットのアプリケーション オブジェクト	UNSIGNED32	6401 0F 10h
	04h	第 4 セットのアプリケーション オブジェクト	UNSIGNED32	6401 10 10h
1A06h	00h	TPDO7 マッピングパラメータ、 サポートするサブインデックス 最大数	UNSIGNED8	04h
	01h	第 1 セットのアプリケーション オブジェクト	UNSIGNED32	6401 11 10h
	02h	第 2 セットのアプリケーション オブジェクト	UNSIGNED32	6401 12 10h
	03h	第 3 セットのアプリケーション オブジェクト	UNSIGNED32	6401 13 10h
	04h	第 4 セットのアプリケーション オブジェクト	UNSIGNED32	6401 14 10h
1A07h	00h	TPDO8 マッピングパラメータ、 サポートするサブインデックス 最大数	UNSIGNED8	04h
	01h	第 1 セットのアプリケーション オブジェクト	UNSIGNED32	6401 15 10h
	02h	第 2 セットのアプリケーション オブジェクト	UNSIGNED32	6401 16 10h
	03h	第 3 セットのアプリケーション オブジェクト	UNSIGNED32	6401 17 10h
	04h	第 4 セットのアプリケーション オブジェクト	UNSIGNED32	6401 18 10h

アプリケーション関連パラメータ(6000h~9FFFh):

インデックス	サブインデックス	記述	タイプ	デフォルト
6206h	01h-40h	デジタル出力点(1~512)エラー モード	UNSIGNED8	FFh
6207h	01h-40h	デジタル出力点(1~512)エラー 発生時の出力値	UNSIGNED8	00h

6443h	01h-40h	アナログ出力点(1~64)エラーモード	UNSIGNED8	01h
6444h	01h-40h	アナログ出力点(1~64) エラー発生時の出力値	INTEGER32	0000 0000h

6.1.12 【1014h】 COB-ID EMCY

DS301 によりますと、EMCY 用の COB-ID を設定する前、Bit31 を 1(無効)に設定する必要があります、さもないと、COB-ID は設定されません。

Bit31	Bit30	Bit11	Bit10	Bit0
0/1 (有効にする/ 無効にする)	予約済み(0)		COB-ID	

6.1.13 【1015h】 抑制時間 EMCY

EMCY の禁制時間で、2 個の EMCY メッセージ間の送信間隔は禁制時間(unit: 100μs)より大きい必要があります。

0 は本機能を無効にする意味です。

6.1.14 【1016h】 Consumer Heartbeat Time

Consumer heartbeat time は装置が Heartbeat メッセージを受信する時間間隔を指定することに用いられます。もし heartbeat time を 0、または Node-ID を 0 もしくは 127 より大きく設定した場合、Consumer heartbeat 機能が無効されます。heartbeat time の単位は 1ms です。

サブインデックス 00h: サブインデックス数

サブインデックス 01h: Consumer heartbeat time

31	24	23	16	15	0
予約済み (00 _h)		Node-ID		Heartbeat time	
MSB			LSB		

6.1.15 【1017h】 Producer Heartbeat Time

Producer Heartbeat Time はノードが heartbeat メッセージ(0x700+Node-ID)を送信する時間間隔です。時間単位 1ms で、デフォルトは 0 で、本機能が無効されていると示しています。

6.1.16 【1029h】 エラー行為オブジェクト

以下の通信エラーが発生する場合：

- CAN Bus が閉じられた
- Life Guarding イベントがトリガーされた
- Heartbeat イベントがトリガーされた

本オブジェクトを用いて、以下のコマンドを出すことができます:

Error Behavior Object		
インデックス	サブインデックス	記述

	クス	
1029h	01h	0 : NMT を前操作状態に変更する(現在、NMT 状態は操作可能状態である場合) 1 : NMT 状態を変更する 2 : NMT を停止状態に変更する

6.2 PDO 通信パラメータ

6.2.1 RxPDO 通信パラメータ

インデックス	サブインデックス	記述	タイプ	属性	デフォルト
1400h	00h	RPDO1 内のサブインデックス数	UNSIGNED8	ro	02h
	01h	RPDO1 の COB-ID	UNSIGNED32	rw	200h + Node-ID
	02h	RPDO1 の送信タイプ	UNSIGNED8	ro	FFh
1401h	00h	RPDO2 内のサブインデックス数	UNSIGNED8	ro	02h
	01h	RPDO2 の COB-ID	UNSIGNED32	rw	300h + Node-ID
	02h	RPDO2 の送信タイプ	UNSIGNED8	ro	FFh
1402h	00h	RPDO3 内のサブインデックス数	UNSIGNED8	ro	02h
	01h	RPDO3 の COB-ID	UNSIGNED32	rw	400h + Node-ID
	02h	RPDO3 の送信タイプ	UNSIGNED8	ro	FFh
1403h	00h	RPDO4 内のサブインデックス数	UNSIGNED8	ro	02h
	01h	RPDO4 の COB-ID	UNSIGNED32	rw	500h + Node-ID
	02h	RPDO4 の送信タイプ	UNSIGNED8	ro	FFh
1404h	00h	RPDO5 内のサブインデックス数	UNSIGNED8	ro	02h
	01h	RPDO5 の COB-ID	UNSIGNED32	rw	8000000
	02h	RPDO5 の送信タイプ	UNSIGNED8	ro	FFh
1405h	00h	RPDO6 内のサブインデックス数	UNSIGNED8	ro	02h
	01h	RPDO6 の COB-ID	UNSIGNED32	rw	8000000
	02h	RPDO6 の送信タイプ	UNSIGNED8	ro	FFh
1406h	00h	RPDO7 内のサブインデックス数	UNSIGNED8	ro	02h
	01h	RPDO7 の COB-ID	UNSIGNED32	rw	8000000
	02h	RPDO7 の送信タイプ	UNSIGNED8	ro	FFh
1407h	00h	RPDO8 内のサブインデックス数	UNSIGNED8	ro	02h
	01h	RPDO8 の COB-ID	UNSIGNED32	rw	8000000
	02h	RPDO8 の送信タイプ	UNSIGNED8	ro	FFh

- サブインデックス 0x01:RPDO の COB-ID

RxPDO の送信は NMT state Operational でのみ実行できます。下記に示した通り、

Bit31 を 1 に設定すると、本機能が無効されます。各 Node-ID は 4 個の RxPDO をサポートし、4 個以上を使用したい場合、使用可能なノード ID を見つけてください。

Bit31	Bit30	Bit11	Bit10	Bit0
0/1 (有効にする/無効にする)	予約済み(0)		COB-ID	

● サブインデックス 02h: RPDO 送信タイプ

00~F0: 同期、FEh/FFh: イベント駆動

数値	記述
00 _h	同期
.....
F0 _h	同期
F1 _h	予約済み
.....
FD _h	同期された
FE _h	イベント駆動(メーカー定義)
FF _h	イベント駆動(標準文書定義)

6.2.2 TxPDO 通信パラメータ

インデックス	サブインデックス	記述	タイプ	属性	デフォルト
1800h	00h	TPDO1 サブインデックス数	UNSIGNED8	ro	05h
	01h	TPDO1 の COB-ID	UNSIGNED32	rw	180h + Node-ID
	02h	TPDO1 の送信タイプ	UNSIGNED8	ro	FFh
	03h	TPDO1 の禁制時間	UNSIGNED16	rw	0
	05h	TPDO1 のイベントタイマー	UNSIGNED16	rw	0
1801h	00h	TPDO2 サブインデックス数	UNSIGNED8	ro	05h
	01h	TPDO2 の COB-ID	UNSIGNED32	rw	280h + Node-ID
	02h	TPDO2 の送信タイプ	UNSIGNED8	ro	FFh
	03h	TPDO2 の禁制時間	UNSIGNED16	rw	0
	05h	TPDO2 の禁制時間	UNSIGNED16	rw	0
1802h	00h	TPDO3 サブインデックス数	UNSIGNED8	ro	05h
	01h	TPDO3 の COB-ID	UNSIGNED32	rw	380h + Node-ID
	02h	TPDO3 の送信タイプ	UNSIGNED8	ro	FFh
	03h	TPDO3 の禁制時間	UNSIGNED16	rw	0
	05h	TPDO3 のイベントタイマー	UNSIGNED16	rw	0
1803h	00h	TPDO4 サブインデックス数	UNSIGNED8	ro	05h

	01h	TPDO4 の COB-ID	UNSIGNED32	rw	480h + Node-ID
	02h	TPDO4 の送信タイプ	UNSIGNED8	ro	FFh
	03h	TPDO4 の禁制時間	UNSIGNED16	rw	0
	05h	TPDO4 のイベントタイマー	UNSIGNED16	rw	0
1804h	00h	TPDO5 サブインデックス数	UNSIGNED8	ro	05h
	01h	TPDO5 の COB-ID	UNSIGNED32	rw	8000000
	02h	TPDO5 の送信タイプ	UNSIGNED8	ro	FFh
	03h	TPDO5 の禁制時間	UNSIGNED16	rw	0
	05h	TPDO5 のイベントタイマー	UNSIGNED16	rw	0
1805h	00h	TPDO6 サブインデックス数	UNSIGNED8	ro	05h
	01h	TPDO6 の COB-ID	UNSIGNED32	rw	8000000
	02h	TPDO6 の送信タイプ	UNSIGNED8	ro	FFh
	03h	TPDO6 の禁制時間	UNSIGNED16	rw	0
	05h	TPDO6 のイベントタイマー	UNSIGNED16	rw	0
1806h	00h	TPDO7 サブインデックス数	UNSIGNED8	ro	05h
	01h	TPDO7 の COB-ID	UNSIGNED32	rw	8000000
	02h	TPDO7 の送信タイプ	UNSIGNED8	ro	FFh
	03h	TPDO7 の禁制時間	UNSIGNED16	rw	0
	05h	TPDO7 のイベントタイマー	UNSIGNED16	rw	0
1807h	00h	TPDO8 サブインデックス数	UNSIGNED8	ro	05h
	01h	TPDO8 の COB-ID	UNSIGNED32	rw	8000000
	02h	Transmission type of TPDO8 の送信タイプ	UNSIGNED8	ro	FFh
	03h	TPDO8 の禁制時間	UNSIGNED16	rw	0
	05h	TPDO8 のイベントタイマー	UNSIGNED16	rw	0

● サブインデックス 01h: TPDO の COB-ID

TxPDO の送信は NMT state Operational でのみ実行できます。下記に示した通り、Bit31 を 1 に設定すると、本機能が無効されます。各 Node-ID は 4 個の TxPDO をサポートし、4 個以上を使用したい場合、使用可能なノード ID を見つけてください。

Bit31	Bit30	Bit11	Bit10	Bit0
0/1 (有効にする/無効にする)	予約済み(0)		COB-ID	

● サブインデックス 02h: TPDO の送信タイプ

00~F0: 同期に用いられ、受信した SYNC 数が指定の数に達した場合、PDO を送信

します。

0xFC: RTR を受信した後、PDO を送信します。

0xFD: RTR を受信し、イベントをトリガーした後、PDO を送信します。

0xFE/0xFF: イベントがトリガーされた後、PDO を送信します。

数値	記述
00 _h	同期 (非周期的)
01 _h	同期 (毎回のsyncコマンドごとに実行する)
02 _h	同期 (毎2回のsyncコマンドごとに実行する)
03 _h	同期 (毎3回のsyncコマンドごとに実行する)
04 _h	同期 (毎4回のsyncコマンドごとに実行する)
.....
F0 _h	同期 (毎240回のsyncコマンドごとに実行する)
F1 _h	reserved予約済み
.....
FB _h	予約済み
FC _h	リモート送信リクエスト (同期)
FD _h	リモート送信リクエスト (イベント駆動)
FE _h	イベント駆動 (メーカー定義)
FF _h	イベント駆動 (標準文書定義)

- サブインデックス 03h: Inhibit Time 抑制時間

PDO の送信タイプが FEh/FFh に設定された場合、抑制時間は PDO 送信の時間間隔を設定することに用いられます。単位は 100μs です。0 は本機能が無効にすることを指しています。

- サブインデックス 05h: Event Timer イベントタイマー

PDO の送信タイプが FEh/FFh に設定された場合、イベントタイマーは PDO 送信の時間間隔を設定することに用いられます。単位は 1ms です。0 は本機能が無効にすることを指しています。

6.2.3 PDO マッピングパラメータ

インデックス	サブインデックス	記述	タイプ	属性	デフォルト
RxPDO1 マッピングパラメータ (デフォルトデジタル出力)					
1600h	00h	RxPDO1 マッピングパラメータ、サポートするサブインデックス最大数	UNSIGNED8	rw	08h
	01h	第1セットのアプリケーションオブジェクト	UNSIGNED32	rw	6200 01 08h
	02h	第2セットのアプリケーションオブジェクト	UNSIGNED32	rw	6200 02 08h

	03h	第 3 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6200 03 08h
	04h	第 4 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6200 04 08h
	05h	第 5 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6200 05 08h
	06h	第 6 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6200 06 08h
	07h	第 7 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6200 07 08h
	08h	第 8 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6200 08 08h
RxPDO2 マッピングパラメータ (デフォルトアナログ出力)					
1601h	00h	RxPDO2 マッピングパラメータ、サポートするサブインデックス最大数	UNSIGNED8	rw	04h
	01h	第 1 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6411 01 10h
	02h	第 2 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6411 02 10h
	03h	第 3 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6411 03 10h
	04h	第 4 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6411 04 10h
RxPDO3 マッピングパラメータ (デフォルト付加アナログ出力)					
1602h	00h	RxPDO3 マッピングパラメータ、サポートするサブインデックス最大数	UNSIGNED8	rw	04h
	01h	第 1 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6411 05 10h
	02h	第 2 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6411 06 10h
	03h	第 3 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6411 07 10h
	04h	第 4 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6411 08 10h
RxPDO4 マッピングパラメータ (付加アナログ出力)					
1603h	00h	RxPDO4 マッピングパラメータ、サポートするサブインデックス最大数	UNSIGNED8	rw	04h

	01h	第 1 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6411 09 10h
	02h	第 2 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6411 0A 10h
	03h	第 3 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6411 0B 10h
	04h	第 4 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6411 0C 10h
RxPDO5 マッピングパラメータ (デフォルトデジタル出力)					
1605h	00h	RxPDO5 マッピングパラメータ、サポートするサブインデックス最大数	UNSIGNED8	rw	08h
	01h	第 1 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6200 09 08h
	02h	第 2 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6200 0A 08h
	03h	第 3 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6200 0B 08h
	04h	第 4 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6200 0C 08h
	05h	第 5 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6200 0D 08h
	06h	第 6 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6200 0E 08h
	07h	第 7 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6200 0F 08h
	08h	第 8 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6200 10 08h
RxPDO6 マッピングパラメータ (デフォルト付加アナログ出力)					
1606h	00h	RxPDO6 マッピングパラメータ、サポートするサブインデックス最大数	UNSIGNED8	rw	04h
	01h	第 1 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6411 0D 10h
	02h	第 2 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6411 0E 10h
	03h	第 3 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6411 0F 10h
	04h	第 4 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6411 10 10h

		エクト			
RxPDO7 マッピングパラメータ (デフォルト付加アナログ出力)					
1607h	00h	RxPDO7 マッピングパラメータ、サポートするサブインデックス最大数	UNSIGNED8	rw	04h
	01h	第 1 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6411 11 10h
	02h	第 2 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6411 12 10h
	03h	第 3 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6411 13 10h
	04h	第 4 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6411 14 10h
RxPDO8 マッピングパラメータ (デフォルト付加アナログ出力)					
1608h	00h	RxPDO7 マッピングパラメータ、サポートするサブインデックス最大数	UNSIGNED8	rw	04h
	01h	第 1 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6411 15 10h
	02h	第 2 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6411 16 10h
	03h	第 3 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6411 17 10h
	04h	第 4 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6411 18 10h
TxPDO 通信パラメータ					
1800h	01h	TPDO1-COB-ID	UNSIGNED32	rw	180h + Node-ID
1801h	01h	TPDO2-COB-ID	UNSIGNED32	rw	280h + Node-ID
1802h	01h	TPDO3-COB-ID	UNSIGNED32	rw	380h + Node-ID
1803h	01h	TPDO4-COB-ID	UNSIGNED32	rw	480h + Node-ID
1804h	01h	TPDO5-COB-ID	UNSIGNED32	rw	0x8000000
1805h	01h	TPDO6-COB-ID	UNSIGNED32	rw	0x8000000
1806h	01h	TPDO7-COB-ID	UNSIGNED32	rw	0x8000000
1807h	01h	TPDO8-COB-ID	UNSIGNED32	rw	0x8000000

TxPDO1 マッピングパラメータ (デジタル入力)					
1A00h	00h	TxPDO1 マッピングパラメータ、サポートするサブインデックス最大数	UNSIGNED8	rw	08h
	01h	第 1 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6000 01 08h
	02h	第 2 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6000 02 08h
	03h	第 3 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6000 03 08h
	04h	第 4 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6000 04 08h
	05h	第 5 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6000 05 08h
	06h	第 6 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6000 06 08h
	07h	第 7 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6000 07 08h
	08h	第 8 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6000 08 08h
TxPDO2 マッピングパラメータ (アナログ入力)					
1A01h	00h	TxPDO1 マッピングパラメータ、サポートするサブインデックス最大数	UNSIGNED8	rw	04h
	01h	第 1 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6401 01 10h
	02h	第 2 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6401 02 10h
	03h	第 3 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6401 03 10h
	04h	第 4 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6401 04 10h
TxPDO3 マッピングパラメータ (付加アナログ入力)					
1A02h	00h	TxPDO3 マッピングパラメータ、サポートするサブインデックス最大数	UNSIGNED8	rw	04h
	01h	第 1 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6401 05 10h
	02h	第 2 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6401 06 10h

	03h	第3セットのアプリケーションオブジェクト	UNSIGNED32	rw	6401 07 10h
	04h	第4セットのアプリケーションオブジェクト	UNSIGNED32	rw	6401 08 10h
TxPDO4 マッピングパラメータ (付加アナログ入力)					
1A03h	00h	TxPDO4 マッピングパラメータ、サポートするサブインデックス最大数	UNSIGNED8	rw	04h
	01h	第1セットのアプリケーションオブジェクト	UNSIGNED32	rw	6401 09 10h
	02h	第2セットのアプリケーションオブジェクト	UNSIGNED32	rw	6401 0A 10h
	03h	第3セットのアプリケーションオブジェクト	UNSIGNED32	rw	6401 0B 10h
	04h	第4セットのアプリケーションオブジェクト	UNSIGNED32	rw	6401 0C 10h
TxPDO5 マッピングパラメータ (デジタル入力)					
1A04h	00h	TxPDO5 マッピングパラメータ、サポートするサブインデックス最大数	UNSIGNED8	rw	08h
	01h	第1セットのアプリケーションオブジェクト	UNSIGNED32	rw	6000 09 08h
	02h	第2セットのアプリケーションオブジェクト	UNSIGNED32	rw	6000 0A 08h
	03h	第3セットのアプリケーションオブジェクト	UNSIGNED32	rw	6000 0B 08h
	04h	第4セットのアプリケーションオブジェクト	UNSIGNED32	rw	6000 0C 08h
	05h	第5セットのアプリケーションオブジェクト	UNSIGNED32	rw	6000 0D 08h
	06h	第6セットのアプリケーションオブジェクト	UNSIGNED32	rw	6000 0E 08h
	07h	第7セットのアプリケーションオブジェクト	UNSIGNED32	rw	6000 0F 08h
	08h	第8セットのアプリケーションオブジェクト	UNSIGNED32	rw	6000 10 08h
TxPDO6 マッピングパラメータ (付加アナログ出力)					
1A05h	00h	TxPDO6 マッピングパラメータ、サポートするサブインデックス最大数	UNSIGNED8	rw	04h

	01h	第 1 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6401 0D 10h
	02h	第 2 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6401 0E 10h
	03h	第 3 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6401 0F 10h
	04h	第 4 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6401 10 10h
TxPDO7 マッピングパラメータ (付加アナログ出力)					
1A06h	00h	TxPDO7 マッピングパラメータ、サポートするサブインデックス最大数	UNSIGNED8	rw	04h
	01h	第 1 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6401 11 10h
	02h	第 2 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6401 12 10h
	03h	第 3 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6401 13 10h
	04h	第 4 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6401 14 10h
TxPDO8 マッピングパラメータ (付加アナログ出力)					
1A07h	00h	TxPDO8 マッピングパラメータ、サポートするサブインデックス最大数	UNSIGNED8	rw	04h
	01h	第 1 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6401 15 10h
	02h	第 2 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6401 16 10h
	03h	第 3 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6401 17 10h
	04h	第 4 セットのアプリケーションオブジェクト	UNSIGNED32	rw	6401 18 10h

6.2.4 マッピングパラメータ

サブインデックス 0x00:

数値	記述
00 _h	マッピングが無効された
01 _h	サブインデックス01 _h が有効された
....
08 _h	サブインデックス01 _h が有効された

サブインデックス 0x01~0x08:

31	16	15	8	7	0
インデックス		サブインデックス			長さ
MSB			LSB		

6.3 I/O 装置オブジェクト辞書エリア(DS401)

6.3.1 デジタル入力点を読み取る

インデックス	サブインデックス	記述	タイプ	属性	デフォルト
6000h		デジタル入力点を読み取る (8-bit)			
	00h	サブインデックス数量	UNSIGNED8	ro	40h
	01h	デジタル入力点 001h to 008h	UNSIGNED8	ro	----
			
	40h	デジタル入力点 1F8h to 200h	UNSIGNED8	ro	----
6020h		デジタル入力点を読み取る (1-bit)			
	00h	サブインデックス数量	UNSIGNED8	ro	80h
	01~80h	デジタル入力点 001h~080h	UNSIGNED8	ro	0
6021h		デジタル入力点を読み取る (1-bit)			
	00h	サブインデックス数量	UNSIGNED8	ro	80h
	01~80h	デジタル入力点 081h~0FFh	UNSIGNED8	ro	0
6021h		デジタル入力点を読み取る (1-bit)			
	00h	サブインデックス数量	UNSIGNED8	ro	80h
	01h~80h	デジタル入力点 100h~180h	UNSIGNED8	ro	0
6022h		デジタル入力点を読み取る (1-bit)			
	00h	サブインデックス数量	UNSIGNED8	ro	80h

	01h~80h	デジタル入力点 181h~1FFh	UNSIGNED8	ro	0
6100h		デジタル入力点を読み取る (16-bit)			
	00h	サブインデックス数量	UNSIGNED8	ro	20h
	01h	デジタル入力点 001h to 010h	UNSIGNED16	ro	----
			
	20h	デジタル入力点 1F0h to 200h	UNSIGNED16	ro	----
6120h		デジタル入力点を読み取る (32-bit)			
	00h	サブインデックス数量	UNSIGNED8	ro	10h
	01h	デジタル入力点 001h to 020h	UNSIGNED32	ro	----
			
	10h	デジタル入力点 1E0h to 200h	UNSIGNED32	ro	----

6.3.2 デジタル出力点を書き込む

インデックス	サブインデックス	記述	タイプ	属性	デフォルト
6200h		デジタル出力点を書き込む (8-bit)			
	00h	サブインデックス数量	UNSIGNED8	ro	40h
	01h	デジタル出力点 001h to 008h	UNSIGNED8	ro	----
			
	40h	デジタル出力点 1F8h to 200h	UNSIGNED8	ro	0
6220h		デジタル出力点を書き込む (1-bit)			
	00h	サブインデックス数量	UNSIGNED8	ro	80h
	01h~80h	デジタル出力点 001h~080h	UNSIGNED8	ro	0
6221h		デジタル出力点を書き込む (1-bit)			
	00h	サブインデックス数量	UNSIGNED8	ro	80h
	01h~80h	デジタル出力点 081h~0FFh	UNSIGNED8	ro	0
6222h		デジタル出力点を書き込む (1-bit)			
	00h	サブインデックス数量	UNSIGNED8	ro	80h
	01h~80h	デジタル出力点 100h~180h	UNSIGNED8	ro	0
6223h		デジタル出力点を書き込む (1-bit)			

	00h	サブインデックス数量	UNSIGNED8	ro	80h
	01h~80h	デジタル出力点 181h~1FFh	UNSIGNED8	ro	0
6300h		デジタル出力点を書き込む (16-bit)			
	00h	サブインデックス数量	UNSIGNED8	ro	20h
	01h	デジタル出力点 001h to 010h	UNSIGNED16	ro	0
			
	20h	デジタル出力点 1F0h to 200h	UNSIGNED16	ro	0
6320h		デジタル出力点を書き込む (32-bit)			
	00h	サブインデックス数量	UNSIGNED8	ro	10h
	01h	デジタル出力点 001h to 020h	UNSIGNED32	ro	0
			
	10h	デジタル出力点 1E0h to 200h	UNSIGNED32	ro	0

6.3.3 アナログ入力チャンネルを読み取る

インデックス	サブインデックス	記述	タイプ	属性	デフォルト
6401h		アナログ入力チャンネルを読み取る(16-bit)			
	00h	サブインデックス数量	UNSIGNED8	ro	40h
	01h	アナログ入力チャンネル 1 内のデジタル値	INTEGER16	ro	0
			
	40h	アナログ入力チャンネル 64 内のデジタル値	INTEGER16	ro	0

6.3.4 アナログ出力チャンネルを書き込む (64 channel)

インデックス	サブインデックス	記述	タイプ	属性	デフォルト
6411h		アナログ出力チャンネルを書き込む(16-bit)			
	00h	サブインデックス数量	UNSIGNED8	ro	40h
	01h	アナログ出力チャンネル 1 内のデジタル値	INTEGER16	ro	0
			

	40h	アナログ出力チャンネル 40 内の デジタル値	INTEGER16	ro	0
--	-----	----------------------------	-----------	----	---

6.4 メーカー特定プロフィールエリア

6.4.1 2000h-2001h : デジタル入力フィルタ

インデックス	サブインデックス	記述	タイプ	属性	デフォルト
2000h	00h	サブインデックス数量	UNSIGNED8	ro	FFh
	01h-FFh	デジタル入力点 1-255 フィルタ 時間	UNSIGNED16	rw	0000h
2001h	00h	サブインデックス数量	UNSIGNED8	ro	FFh
	0h1-FFh	デジタル入力点 256-512 フィル タ時間	UNSIGNED16	rw	0000h

6.4.2 3000h~300Fh: モジュールレジスタ

インデックス	記述
3000h	第 1 台のモジュールレジスタを読み取る/書き込む
3001h	第 2 台のモジュールレジスタを読み取る/書き込む
3002h	第 3 台のモジュールレジスタを読み取る/書き込む
3003h	第 4 台のモジュールレジスタを読み取る/書き込む
3004h	第 5 台のモジュールレジスタを読み取る/書き込む
3005h	第 6 台のモジュールレジスタを読み取る/書き込む
3006h	第 7 台のモジュールレジスタを読み取る/書き込む
.....
300Fh	第 16 台のモジュールレジスタを読み取る/書き込む

サブインデックス	記述	タイプ	属性	デフォルト
00h	サブインデックス数量	UNSIGNED8	ro	80h
01h	モジュールレジスタアドレス 0	UNSIGNED16	rw	
02h	モジュールレジスタアドレス 1	UNSIGNED16	rw	
03h	モジュールレジスタアドレス 2	UNSIGNED16		
04h	モジュールレジスタアドレス 3	UNSIGNED16	rw	
.....
80h	モジュールレジスタアドレス 127	UNSIGNED16	rw	

6.4.3 4000h-4007h : iBus 及びモジュール情報

インデックス	サブインデックス	記述	タイプ	属性	デフォルト
4000h	00h	接続されたモジュール台数	UNSIGNED16	ro	----
	01h	予約済み	UNSIGNED16	ro	----
	02h	デジタル入力点数	UNSIGNED16	ro	----
	03h	デジタル出力点数	UNSIGNED16	ro	----
	04h	アナログ入力チャンネル数	UNSIGNED16	ro	----
	05h	アナログ出力チャンネル数	UNSIGNED16	ro	----
	06h	現在の総消費電力	UNSIGNED16	ro	----
	07h	サポートできる消費電力	UNSIGNED16	ro	----
4001h	01h	第 1 台モジュールのファームウェアバージョン	UNSIGNED16	ro	----
	02h	第 2 台モジュールのファームウェアバージョン	UNSIGNED16	ro	----

	10h	第 16 台モジュールのファームウェアバージョン	UNSIGNED16	ro	----
4002h	01h	第 1 台モジュールのハードウェアバージョン	UNSIGNED16	ro	----
	02h	第 2 台モジュールのハードウェアバージョン	UNSIGNED16	ro	----

	10h	第 16 台モジュールのハードウェアバージョン	UNSIGNED16	ro	----
4003h	01h	第 1 台モジュールの消費電力	UNSIGNED16	ro	----
	02h	第 2 台モジュールの消費電力	UNSIGNED16	ro	----

	10h	第 16 台モジュールの消費電力	UNSIGNED16	ro	----
4004h	01h	第 1 台モジュールのデジタル入力点数	UNSIGNED16	ro	----
	02h	第 2 台モジュールのデジタル入力点数	UNSIGNED16	ro	----

	10h	第 16 台モジュールのデジタル入力点数	UNSIGNED16	ro	----
4005h	01h	第 1 台モジュールのデジタル出力点数	UNSIGNED16	ro	
	02h	第 2 台モジュールのデジタル出力点数	UNSIGNED16	ro	

	
	10h	第 16 台モジュールのデジタル出力点数	UNSIGNED16	ro	
4006h	01h	第 1 台モジュールのアナログ入力チャンネル数	UNSIGNED16	ro	
	02h	第 2 台モジュールのアナログ入力チャンネル数	UNSIGNED16	ro	
	
	10h	第 16 台モジュールのアナログ入力チャンネル数	UNSIGNED16	ro	
4007h	01h	第 1 台モジュールのアナログ出力チャンネル数	UNSIGNED16	ro	
	02h	第 2 台モジュールのアナログ出力チャンネル数	UNSIGNED16	ro	
	
	10h	第 16 台モジュールのアナログ出力チャンネル数	UNSIGNED16	ro	

7. 消費電力

I/O モジュールはシステム電源と I/O 外部ドライブ電源を使用し、両者を分けて計算しています。カプラと I/O モジュール内部はシステム電源を使用し、I/O の切り替え及び駆動は I/O 外部ドライブ電源を使用します。なので、カプラの消費電力を計算する場合、I/O モジュールのシステム電源だけを計算すればいいです。複数のモジュールを接続する場合、消費電力をご留意ください。以下はカプラと I/O モジュールの消費電力(参考用)です：

タイプ	装置名	消費電力(5V)	入力電圧(5V)
カプラ	iR-ETN	220mA/1.1w	2A/10w
	iR-COP	170mA/0.85w	2A/10w
デジタル I/O モジュール	iR-DM16-P	130mA/0.65w	--
	iR-DM16-N	130mA/0.65w	--
	iR-DQ08-R	220mA/1.1w	--
	iR-DQ16-N	205mA/1.02w	
	iR-DQ16-P	196mA/0.984w	
	iR-DI16-K	83mA/0.418w	
	iR-AQ04-VI	55mA/0.275w	
アナログ I/O モジュール	iR-AI04-VI	70mA/0.35W	
	iR-AM06-VI	70mA/0.35W	
	iR-AI04-TR	65mA/0.325w	
モーションコントロール	iR-PU01-P	108mA/0.54W	

以下は計算例です:

タイプ	装置名	消費電力	入力電圧
カブラ	iR-COP	170mA/0.85w	2A/10w
モジュール	iR-DQ08-R *8	220mA*8=1.76A	X
システム	消費電力: 170mA + 1.76A = 1.93 A 入力電圧: 2A > 1.93A		

8. CODESYS で Remote I/O に接続する

cMT+CODESYS 及び Remote I/O のマニュアルを参考してください。

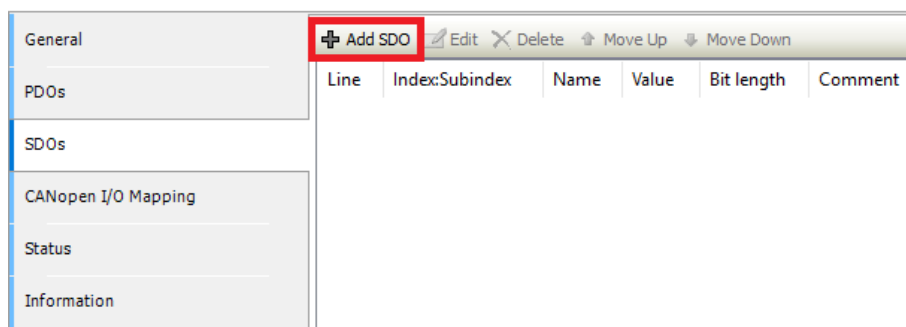
9. CODESYS でアナログモジュールを設定する

iR-COP に接続しているアナログモジュールのパラメータを設定するには、2 種類の方法があります。

1. SDO を追加します。設定を完了してログインすると、パラメータを書き込むことができます。
2. Weintek_Library 機能ブロックを使用します。機能ブロックをトリガーすると、指定したアドレスに読み取る・書き込むことができます。

9.1 SDO を追加する

[iR-COP] » [SDOs] » [Add SDO]



SDO を追加した後、プログラムにログインします。パラメータはログインした後でモジュールに書き込まれます。

9.1.1 アナログ入力パラメータを書き込む

Index = 16#3001 (本例では、第 2 台のモジュールに当たります)

Sub Index = 16#15 (アナログ入力チャンネル 0 のチャンネルモードに関しては、iR-AI04-VI, iR-AM06-VI, iR-AQ04-VI ユーザーマニュアルのモジュールレジスタの章節にご参照ください)

Bit length = 16 (全てのレジスタの長さは 16-bit です)

Value = 1 (数値を書き込みます。iR-AI04-VI, iR-AM06-VI, iR-AQ04-VI ユーザーマニュアルのモジュールレジスタの章節にご参照ください)

Select Item from Object Directory

Index/Subindex	Name	AccessType	Type	Default
16#1003	Pre-defined error field			
16#100C:16#00	Guard Time	RW	UINT	16#00
16#100D:16#00	Life Time Factor	RW	USINT	16#00
16#1010	Store Parameters			
16#1011	Restore default parameters			
16#1014:16#00	COB-ID EMCY	RW	UDINT	\$NODEID+16#80
16#1015:16#00	Emergency Inhibit Time	RW	UINT	16#00
16#1016	Consumer heartbeat time			
16#1017:16#00	Producer heartbeat time	RW	UINT	16#00
16#1029	Error behavior object			
16#1400	RxPDO 1 comm. parameter			
16#1401	RxPDO 2 comm. parameter			
16#1402	RxPDO 3 comm. parameter			
16#1403	RxPDO 4 comm. parameter			
16#1404	RxPDO 5 comm. parameter			
16#1405	RxPDO 6 comm. parameter			
16#1406	RxPDO 7 comm. parameter			

Name: AM06-VI Input Channel 0 Mode

Index: 16#3001 Bit length: 16

SubIndex: 16#15 Value: 1

OK Cancel

Line	Index/Subindex	Name	Value	Bit length
1	16#3001:16#02	AM06-VI Output Channel 1 Mode	1	16

9.1.2 アナログ出力パラメータを書き込む

Index = 16#3001 (本例では、第 2 台のモジュールに当たります)

Sub Index = 16#02 (アナログ出力チャンネル 1 のチャンネルモードに関しては、iR-AI04-VI, iR-AM06-VI, iR-AQ04-VI ユーザーマニュアルのモジュールレジスタの章節にご参照ください)

Bit length = 16 (全てのレジスタの長さは 16-bit です)

Value = 1 (数値を書き込みます。iR-AI04-VI, iR-AM06-VI, iR-AQ04-VI ユーザーマニュアルのモジュールレジスタの章節にご参照ください)

Select Item from Object Directory

Index:Subindex	Name	AccessType	Type	Default
16#1003	Pre-defined error field			
16#100C:16#00	Guard Time	RW	UINT	16#00
16#100D:16#00	Life Time Factor	RW	USINT	16#00
16#1010	Store Parameters			
16#1011	Restore default parameters			
16#1014:16#00	COB-ID EMCY	RW	UDINT	\$NODEID+16#80
16#1015:16#00	Emergency Inhibit Time	RW	UINT	16#00
16#1016	Consumer heartbeat time			
16#1017:16#00	Producer heartbeat time	RW	UINT	16#00
16#1029	Error behavior object			
16#1400	RxPDO 1 comm. parameter			
16#1401	RxPDO 2 comm. parameter			
16#1402	RxPDO 3 comm. parameter			
16#1403	RxPDO 4 comm. parameter			
16#1404	RxPDO 5 comm. parameter			
16#1405	RxPDO 6 comm. parameter			
16#1406	RxPDO 7 comm. parameter			

Name: AM06-VI Output Channel 1 Mode

Index: 16#3001 Bit length: 16

SubIndex: 16#2 Value: 1

OK Cancel

General	+ Add SDO Edit Delete Move Up Move Down				
PDOs	Line	Index:Subindex	Name	Value	Bit length
SDOs	1	16#3001:16#02	AM06-VI Output Channel 1 Mode	1	16
CANopen I/O Mapping					
Status					
Information					

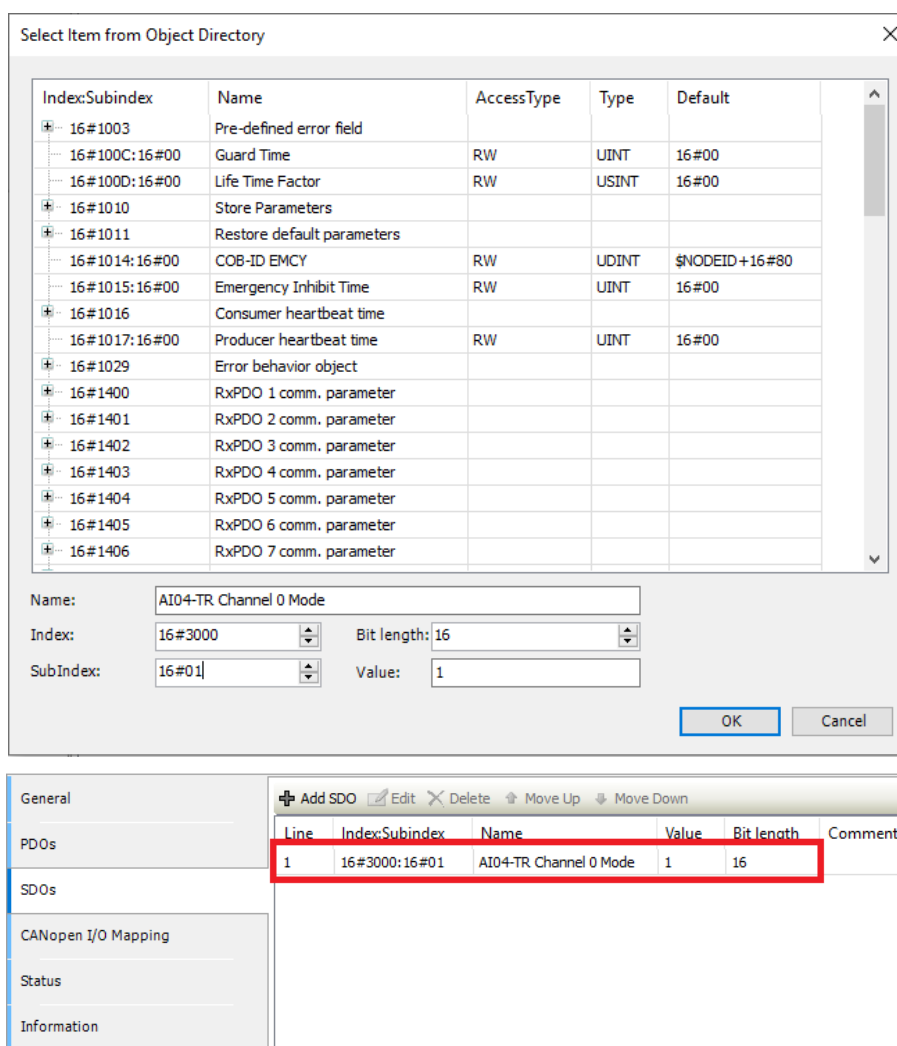
9.1.3 温度入力パラメータを書き込む

Index = 16#3000 (本例では、第 1 台のモジュールに当たります)

Sub Index = 16#01 (温度入力チャンネル 0 のチャンネルモードに関しては、iR-AI04-TR ユーザーマニュアルのモジュールレジスタの章節にご参照ください)

Bit length = 16 (全てのレジスタの長さは 16-bit です)

Value = 1 (数値を書き込みます。iR-AI04-TR ユーザーマニュアルのモジュールレジスタの章節にご参照ください)



9.2 Weintek_Library 機能ブロック

[Library Manager] » [Add library] » [Miscellaneous] » [Weintek CODESYS Library]

AI_Ch_Pa, Analog_Config, Analog_VI_Read, AO_Ch_Pa の 4 個の機能ブロックは、iR-COP に接続しているアナログモジュールにパラメータの読み取り・書き込みを実行できます。

9.2.1 Analog_Config – 1 個のレジスタを読み取り/書き込み

xEnable = 立ち上がりエッジで、xRead_Write, wIndex, bSubIndex, bNode_ID のパラメータに基づいて読み取り・書き込みます。

xRead_Write = 読み取りまたは書き込みます。

wIndex = モジュールのインデックスです。

bSubIndex = レジスタの機能です。iR-AI04-TR ユーザーマニュアル、または iR-AI04-VI, iR-AM06-VI, iR-AQ04-VI ユーザーマニュアルのモジュールレジスタの章節にご参照ください。

bNode_ID = iR-COP のノードです。

xConfirm = 実行が完了したと示しています。

iData = 読み取られ、または書き込まれたデータ内容です。

例：

第 1 台のモジュール(iR-AI04-TR)のチャンネル 0 のチャンネルモードを読み取ります。









宣言

```

1 | PROGRAM PLC_PRG
2 |   VAR
3 |     Analog_Register_Config : weintek.Analog_Config ;
4 |     xEnable : BOOL ;
5 |     xRead_Write_Switch : BOOL;
6 |     wIndex : WORD ;
7 |     bSub_Index : BYTE ;
8 |     bNode_ID : BYTE ;
9 |     xConfirm : BOOL ;
10 |    iRegister_Data : INT ;

```

アクセスするアドレスを入力します。

Device.Application.PLC_PRG			
Expression	Type	Value	Prepared value
 Analog_Register_Config	weintek.Analog_Config		
 xEnable	BOOL	FALSE	
 xRead_Write_Switch	BOOL	FALSE	
 wIndex	WORD	16#0000	16#3000
 bSub_Index	BYTE	16#00	16#01
 bNode_ID	BYTE	16#00	16#01
 xConfirm	BOOL	FALSE	
 iRegister_Data	INT	16#0000	


```

1 | Analog_Register_Config(
2 |   xEnable[FALSE] := xEnable[FALSE],
3 |   xRead_Write[FALSE] := xRead_Write_Switch[FALSE],
4 |   wIndex[16#0000] := wIndex[16#0000 <16#3000>],
5 |   bSubIndex[16#00] := bSub_Index[16#00 <16#01>],
6 |   bNode_ID[16#00] := bNode_ID[16#00 <16#01>],
7 |   xConfirm[FALSE] => xConfirm[FALSE],
8 |   iData[16#0000] := iRegister_Data[16#0000] );RETURN

```

xEnable をトリガーします。読み取った数値が iRegister_Data に表示されます。

Expression	Type	Value
Analog_Register_Config	weintek.Analog_Config	
xEnable	BOOL	TRUE
xRead_Write_Switch	BOOL	FALSE
wIndex	WORD	16#3000
bSub_Index	BYTE	16#01
bNode_ID	BYTE	16#01
xConfirm	BOOL	TRUE
iRegister_Data	INT	16#0001


```

1 Analog_Register_Config(
2     xEnable TRUE := xEnable  TRUE,
3     xRead_Write FALSE := xRead_Write_Switch FALSE,
4     wIndex 16#3000 := wIndex  16#3000,
5     bSubIndex 16#01 := bSub_Index  16#01,
6     bNode_ID 16#01 := bNode_ID  16#01,
7     xConfirm TRUE => xConfirm TRUE,
8     iData 16#0001 := iRegister_Data 16#0001 ); RETURN

```

iRegister_Data 内の数値=1(J-Type)。

※xRead_Write_Switch = TRUE、xEnable をトリガーすると、書き込むことになります。

9.2.2 AI_Ch_Pa – アナログ入力チャンネルをに読み取る・書き込む

xEnable =立ち上がりエッジで、xRead_Write, wIndex, bSubIndx, bNode_ID のパラメータに基づいて読み取り・書き込みます。

xRead_Write = 読み取りまたは書き込みます。

bNode_ID = iR-COP のノードです。

wIndex = モジュールのインデックスです。

bChannel = チャンネルを操作します(チャンネル 0~3)。

xDone =実行が完了したと示しています。

iMax_peak, iMin_peak, iMode, iScale_Max, iScale_Min, iSamp_times = アナログ入力のチャンネルパラメータです。

例：

















第 2 台のモジュール(iR-AM06-VI)のチャンネル 0 の全てのパラメータを読み取る宣言

```




2  VAR
3      AI_Config_Ch : weintek.AI_Ch_Pa ;
4      xEnable : BOOL ;
5      xRead_Write_Switch : BOOL;
6      bNode_ID : BYTE ;
7      wIndex : WORD ;
8      bChannel : BYTE ;
9      xDone : BOOL ;
10     iMax_peak : INT ;
11     iMin_peak : INT ;
12     iMode : INT ;
13     iScale_Max : INT ;
14     iScale_Min : INT ;
15     iFilter_Frame_Size : INT ;

```

アドレスを入力します。

Expression	Type	Value
 AI_Config_Ch	weintek.AI_Ch_Pa	
 xEnable	BOOL	FALSE
 xRead_Write_Switch	BOOL	FALSE
 bNode_ID	BYTE	 16#01
 wIndex	WORD	 16#3001
 bChannel	BYTE	 16#00
 xDone	BOOL	FALSE
 iMax_peak	INT	16#0000
 iMin_peak	INT	16#0000
 iMode	INT	16#0000
 iScale_Max	INT	16#0000
 iScale_Min	INT	16#0000
 iFilter_Frame_Size	INT	16#0000


```

1  AI_Config_Ch(
2      xEnable FALSE := xEnable FALSE,
3      xRead_Write FALSE := xRead_Write_Switch FALSE,
4      bNode_ID 16#01 := bNode_ID  16#01,
5      bChannel 16#00 := bChannel  16#00,
6      wIndex 16#3001 := wIndex  16#3001,
7      xDone FALSE => xDone FALSE,
8      iMax_peak 16#0000 => iMax_peak 16#0000,
9      iMin_peak 16#0000 => iMin_peak 16#0000,
10     iMode 16#0000 := iMode 16#0000,
11     iScale_Max 16#0000 := iScale_Max 16#0000,
12     iScale_Min 16#0000 := iScale_Min 16#0000,
13     iSamp_times 16#0000 := iFilter_Frame_Size 16#0000 );
14     RETURN

```

xEnable をトリガーし、チャンネルパラメータを読み取ります。

Expression	Type	Value
AI_Config_Ch	weintek.AI_Ch_Pa	
xEnable	BOOL	TRUE
xRead_Write_Switch	BOOL	FALSE
bNode_ID	BYTE	16#01
wIndex	WORD	16#3001
bChannel	BYTE	16#00
xDone	BOOL	TRUE
iMax_peak	INT	16#0000
iMin_peak	INT	16#FFA8
iMode	INT	16#0001
iScale_Max	INT	16#7D00
iScale_Min	INT	16#8300
iFilter_Frame_Size	INT	16#0005


```

1  AI_Config_Ch (
2      xEnable TRUE := xEnable TRUE,
3      xRead_Write FALSE := xRead_Write_Switch FALSE,
4      bNode_ID 16#01 := bNode_ID 16#01,
5      bChannel 16#00 := bChannel 16#00,
6      wIndex 16#3001 := wIndex 16#3001,
7      xDone TRUE => xDone TRUE,
8      iMax_peak 16#0000 => iMax_peak 16#0000,
9      iMin_peak 16#FFA8 => iMin_peak 16#FFA8,
10     iMode 16#0001 := iMode 16#0001,
11     iScale_Max 16#7D00 := iScale_Max 16#7D00,
12     iScale_Min 16#8300 := iScale_Min 16#8300,
13     iSamp_times 16#0005 := iFilter_Frame_Size 16#0005 );
14  RETURN

```

チャンネルパラメータが読み取られました。

※xRead_Write_Switch = TRUE、xEnable をトリガーすると、書き込むことになります。

9.2.3 AO_Ch_Pa – アナログ出力のチャンネルに読み取る・書き込む

xEnable = 立ち上がりエッジで、xRead_Write, wIndex, bSubIndx, bNode_ID パラメータに基づいて読み取り・書き込みます。

xRead_Write = 読み取りまたは書き込みます。

bNode_ID = iR-COP のノードです。

wIndex = モジュールのインデックスです。

bChannel = チャンネルを操作します(チャンネル 0~3)。

xDone = 実行が完了したと示しています。

iMode, iScale_Max, iScale_Min, iUpdate_times = アナログ出力のチャンネルパラメータです。

例：















第2台のモジュール(iR-AM06-VI)のチャンネル0の全てのパラメータを読み取ります。
宣言

```




1  PROGRAM PLC_PRG
2  VAR
3      AQ_Config_Ch : weintek.AO_Ch_Pa ;
4      xEnable : BOOL ;
5      xRead_Write_Switch : BOOL;
6      bNode_ID : BYTE ;
7      wIndex : WORD ;
8      bChannel : BYTE ;
9      xDone : BOOL ;
10     iMode : INT ;
11     iScale_Max : INT ;
12     iScale_Min : INT ;
13     iUpdate_time : INT ;

```

アドレスを入力します。
















Expression	Type	Value
 AQ_Config_Ch	weintek.AO_Ch_Pa	
 xEnable	BOOL	FALSE
 xRead_Write_Switch	BOOL	FALSE
 bNode_ID	BYTE	 16#01
 wIndex	WORD	 16#3001
 bChannel	BYTE	 16#01
 xDone	BOOL	FALSE
 iMode	INT	16#0000
 iScale_Max	INT	16#0000
 iScale_Min	INT	16#0000
 iUpdate_time	INT	16#0000

```





1  AQ_Config_Ch(
2      xEnable FALSE := xEnable FALSE,
3      xRead_Write FALSE := xRead_Write_Switch FALSE,
4      bNode_ID 16#01 := bNode_ID  16#01,
5      wIndex 16#3001 := wIndex  16#3001,
6      bChannel 16#01 := bChannel  16#01,
7      xDone FALSE => xDone FALSE,
8      iMode 16#0000 := iMode 16#0000,
9      iScale_Max 16#0000 := iScale_Max 16#0000,
10     iScale_Min 16#0000 := iScale_Min 16#0000,
11     iUpdate_time 16#0000 := iUpdate_time 16#0000 );
12  RETURN

```

xEnable をトリガーし、チャンネルパラメータを読み取ります。

Expression	Type	Value
 AQ_Config_Ch	weintek.AO_Ch_Pa	
 xEnable	BOOL	 TRUE
 xRead_Write_Switch	BOOL	FALSE
 bNode_ID	BYTE	 16#01
 wIndex	WORD	 16#3001
 bChannel	BYTE	 16#01
 xDone	BOOL	TRUE
 iMode	INT	16#0001
 iScale_Max	INT	16#7D00
 iScale_Min	INT	16#8300
 iUpdate_time	INT	16#0000


```

1  AQ_Config_Ch(
2      xEnable TRUE := xEnable  TRUE,
3      xRead_Write FALSE := xRead_Write_Switch FALSE,
4      bNode_ID 16#01 := bNode_ID  16#01,
5      wIndex 16#3001 := wIndex  16#3001,
6      bChannel 16#01 := bChannel  16#01,
7      xDone TRUE => xDone TRUE,
8      iMode 16#0001 := iMode 16#0001,
9      iScale_Max 16#7D00 := iScale_Max 16#7D00,
10     iScale_Min 16#8300 := iScale_Min 16#8300,
11     iUpdate_time 16#0000 := iUpdate_time 16#0000 );
12 RETURN

```

チャンネルパラメータが読み取られました。

※xRead_Write_Switch = TRUE、xEnable をトリガーすると、書き込むことになります。

9.2.4 Analog_VI_Read – アナログモジュールの全てのレジスタを読み取る

xEnable = 立ち上がりエッジで、wIndex、bNode_ID パラメータに基づいて読み取ります。

bNode_ID = iR-COP のノードです。

wIndex = モジュールのインデックスです。

xDone = 実行が完了したと示しています。

aiRegister = アナログモジュールの全てのレジスタです。

例：

第 2 台のモジュール(AM06-VI)のチャンネル 0 の全てのパラメータを読み取ります。

































宣言

```































1  PROGRAM PLC_PRG
2  VAR
3      Analog_Register_Read : weintek.Analog_VI_Read ;
4      xEnable : BOOL ;
5      bNode_ID : BYTE ;
6      wIndex : WORD ;
7      xDone : BOOL ;
8      aiRegister : ARRAY[0..43] OF INT ;

```




アドレスを入力します。

Expression	Type	Value
 Analog_Register_Read	weintek.Analog_VI_...	
 xEnable	BOOL	FALSE
 bNode_ID	BYTE	 16#01
 wIndex	WORD	 16#3000
 xDone	BOOL	FALSE
  aiRegister	ARRAY [0..43] OF INT	
 aiRegister[0]	INT	16#0000
 aiRegister[1]	INT	16#0000
 aiRegister[2]	INT	16#0000
 aiRegister[3]	INT	16#0000
 aiRegister[4]	INT	16#0000
 aiRegister[5]	INT	16#0000
 aiRegister[6]	INT	16#0000
 aiRegister[7]	INT	16#0000
 aiRegister[8]	INT	16#0000
 aiRegister[9]	INT	16#0000
 aiRegister[10]	INT	16#0000
 aiRegister[11]	INT	16#0000
 aiRegister[12]	INT	16#0000
 aiRegister[13]	INT	16#0000
 aiRegister[14]	INT	16#0000
 aiRegister[15]	INT	16#0000
 aiRegister[16]	INT	16#0000
 aiRegister[17]	INT	16#0000
 aiRegister[18]	INT	16#0000
 aiRegister[19]	INT	16#0000
 aiRegister[20]	INT	16#0000
<pre> 1 Analog_Register_Read(2 xEnable FALSE := xEnable FALSE, 3 bNode_ID 16#01 := bNode_ID  16#01, 4 wIndex 16#3000 := wIndex  16#3000, 5 xDone FALSE => xDone FALSE, 6 aiRegister => aiRegister); </pre>		

xEnable をトリガーし、チャンネルパラメータを読み取ります。

Expression	Type	Value
 Analog_Register_Read	weintek.Analog_VI_...	
 xEnable	BOOL	 TRUE
 bNode_ID	BYTE	 16#01
 wIndex	WORD	 16#3000
 xDone	BOOL	TRUE
 aiRegister	ARRAY [0..43] OF INT	
 aiRegister[0]	INT	16#0002
 aiRegister[1]	INT	16#0003
 aiRegister[2]	INT	16#0004
 aiRegister[3]	INT	16#0005
 aiRegister[4]	INT	16#7D00
 aiRegister[5]	INT	16#7D00
 aiRegister[6]	INT	16#7D00
 aiRegister[7]	INT	16#7D00
 aiRegister[8]	INT	16#8300
 aiRegister[9]	INT	16#0000
 aiRegister[10]	INT	16#8300
 aiRegister[11]	INT	16#0000
 aiRegister[12]	INT	16#0000
 aiRegister[13]	INT	16#0000
 aiRegister[14]	INT	16#0000
 aiRegister[15]	INT	16#0000
 aiRegister[16]	INT	16#0000
 aiRegister[17]	INT	16#0000
 aiRegister[18]	INT	16#0000
 aiRegister[19]	INT	16#0000
 aiRegister[20]	INT	16#0001


```

1 Analog_Register_Read(
2   xEnable TRUE := xEnable  TRUE,
3   bNode_ID 16#01 := bNode_ID  16#01,
4   wIndex 16#3000 := wIndex  16#3000,
5   xDone TRUE => xDone TRUE,
6   aiRegister=> aiRegister);

```

全てのパラメータが読み取られました。

付録A. iR-COP デザインリソース

《[UM018016J CODESYS iR Resources](#)》に参照してください。