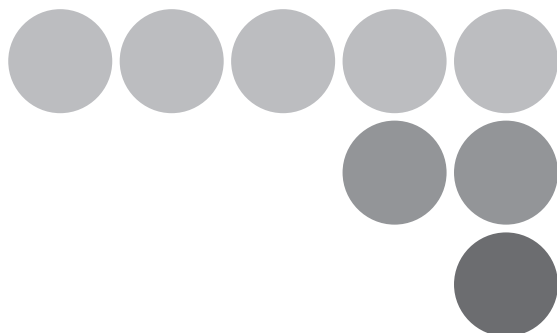




セーフティレーザスキャナ OS32C-DM シリーズ

接続マニュアル



本マニュアルは、セーフティレーザスキャナOS32Cシリーズユーザーズ
マニュアル(P/N 99863-0040またはマニュアル番号SCHG-729)と共に
ご使用ください

EtherNet/IP™

Translation of the original detailed instructions

(本ユーザーズマニュアルは英語詳細版オリジナルからの翻訳です)

はじめに

このたびはセーフティレーザスキャナ形OS32C(以下OS32Cと呼びます)をお買い上げいただき、ありがとうございます。
本書は、形OS32C-□□□-DMをEtherNet/IPで各種機器に接続する手段とその確認方法について説明しています。
本書は、必ず、「セーフティレーザスキャナ OS32Cシリーズ ユーザーズマニュアル(SCHG-729)」(以下OS32Cユーザーズマニュアルと呼びます)と共にご使用ください。
ご使用に際して下記のことを守ってください。

- ・OS32Cは設置される機械について十分に理解されている「責任者」がお取り扱いください。
- ・本書の中で「責任者」とは、機械の設計・設置・運用・保守・廃棄の各段階において、安全確保を行うための資格および権限と責任のある人物のことです。
- ・OS32Cは、設置環境、機械の性能・機能に対応して正しく使用されることが前提となっています。関係者によるリスクアセスメントを実施した上での設置をお願いします。
- ・ご使用の前に本書をよく読んで十分にご理解の上、正しくご使用ください。
- ・本書はいつでも参照できるように、お手元に大切に保管してください。

商標

- ・Windows、Windows XP、Windows 7、Windows 8.1、Windows 10は、米国Microsoft Corporationの米国およびその他の国における登録商標または商標です。
- ・ODVA、EtherNet/IP™、DeviceNet™、ControlNet™、CIP™、CIP Safety™ はODVA, Incの商標です。
- ・その他記載されている会社名、製品名は各社の登録商標または商標です。
- ・スクリーンショットはマイクロソフトの許可を得て使用しています。

ご承諾事項

「当社商品」について特別の合意がない場合には、お客様のご購入先にかかわらず、本ご承諾事項記載の条件を適用いたします。

1. 定義

本ご承諾事項中の用語の定義は次のとおりです。

- (1) 「当社商品」：「当社」のF Aシステム機器、汎用制御機器、センシング機器、電子・機構部品
- (2) 「カタログ等」：「当社商品」に関する、ベスト制御機器オムロン、電子・機構部品総合カタログ、その他のカタログ、仕様書、取扱説明書、マニュアル等であって電磁的方法で提供されるものも含まれます。
- (3) 「利用条件等」：「カタログ等」に記載の、「当社商品」の利用条件、定格、性能、動作環境、取り扱い方法、利用上の注意、禁止事項その他
- (4) 「お客様用途」：「当社商品」のお客様におけるご利用方法であって、お客様が製造する部品、電子基板、機器、設備またはシステム等への「当社商品」の組み込み又は利用を含みます。
- (5) 「適合性等」：「お客様用途」での「当社商品」の(a)適合性、(b)動作、(c)第三者の知的財産の非侵害、(d)法令の遵守および(e)各種規格の遵守

2. 記載事項のご注意

「カタログ等」の記載内容については次の点をご理解ください。

- (1) 定格値および性能値は、単独試験における各条件のもとで得られた値であり、各定格値および性能値の複合条件のもとで得られる値を保証するものではありません。
- (2) 参考データはご参考として提供するもので、その範囲で常に正常に動作することを保証するものではありません。
- (3) 利用事例はご参考ですので、「当社」は「適合性等」について保証いたしかねます。
- (4) 「当社」は、改善や当社都合等により、「当社商品」の生産を中止し、または「当社商品」の仕様を変更することがあります。

3. ご利用にあたってのご注意

ご採用およびご利用に際しては次の点をご理解ください。

- (1) 定格・性能ほか「利用条件等」を遵守しご利用ください。
- (2) お客様ご自身にて「適合性等」をご確認いただき、「当社商品」のご利用の可否をご判断ください。
「当社」は「適合性等」を一切保証いたしかねます。
- (3) 「当社商品」がお客様のシステム全体の中で意図した用途に対して、適切に配電・設置されていることをお客様ご自身で、必ず事前に確認してください。
- (4) 「当社商品」をご使用の際には、(i)定格および性能に対し余裕のある「当社商品」のご利用、冗長設計などの安全設計、(ii)「当社商品」が故障しても、「お客様用途」の危険を最小にする安全設計、(iii)利用者に危険を知らせるための、安全対策のシステム全体としての構築、(iv)「当社商品」および「お客様用途」の定期的な保守、の各事項を実施してください。
- (5) 「当社」はDDoS攻撃（分散型DoS攻撃）、コンピュータウイルスその他の技術的な有害プログラム、不正アクセスにより、「当社商品」、インストールされたソフトウェア、またはすべてのコンピュータ機器、コンピュータプログラム、ネットワーク、データベースが感染したとしても、そのことにより直接または間接的に生じた損失、損害その他の費用について一切責任を負わないものとします。
お客様ご自身にて、①アンチウイルス保護、②データ入出力、③紛失データの復元、④「当社商品」またはインストールされたソフトウェアに対するコンピュータウイルス感染防止、⑤「当社商品」に対する不正アクセス防止についての十分な措置を講じてください。

- (6)「当社商品」は、一般工業製品向けの汎用品として設計製造されています。従いまして、次に掲げる用途での使用は意図しておらず、お客様が「当社商品」をこれらの用途に使用される際には、「当社」は「当社商品」に対して一切保証をいたしません。ただし、次に掲げる用途であっても、「当社」の意図した特別な商品用途の場合や特別の合意がある場合は除きます。
- (a)高い安全性が必要とされる用途（例：原子力制御設備、燃焼設備、航空・宇宙設備、鉄道設備、昇降設備、娯楽設備、医用機器、安全装置、その他生命・身体に危険が及ぶ用途）
 - (b)高い信頼性が必要な用途（例：ガス・水道・電気等の供給システム、24時間連続運転システム、決済システムほか権利・財産を取り扱う用途など）
 - (c)厳しい条件または環境での用途（例：屋外に設置する設備、化学的汚染を被る設備、電磁的妨害を被る設備、振動・衝撃を受ける設備など）
 - (d)「カタログ等」に記載のない条件や環境での用途
- (7) 上記3.(6)(a)から(d)に記載されている他、「本カタログ等記載の商品」は自動車（二輪車含む。以下同じ）向けではありません。自動車に搭載する用途には利用しないで下さい。自動車搭載用商品については当社営業担当者にご相談ください。

4. 保証条件

「当社商品」の保証条件は次のとおりです。

- (1) 保証期間 ご購入後 1 年間といたします。
（ただし「カタログ等」に別途記載がある場合を除きます。）
- (2) 保証内容 故障した「当社商品」について、以下のいずれかを「当社」の任意の判断で実施します。
 - (a) 当社保守サービス拠点における故障した「当社商品」の無償修理
（ただし、電子・機構部品については、修理対応は行いません。）
 - (b) 故障した「当社商品」と同数の代替品の無償提供
- (3) 保証対象外 故障の原因が次のいずれかに該当する場合は、保証いたしません。
 - (a) 「当社商品」本来の使い方以外のご利用
 - (b) 「利用条件等」から外れたご利用
 - (c) 本ご承諾事項「3. ご利用にあたってのご注意」に反するご利用
 - (d) 「当社」以外による改造、修理による場合
 - (e) 「当社」以外の者によるソフトウェアプログラムによる場合
 - (f) 「当社」からの出荷時の科学・技術の水準では予見できなかった原因
 - (g) 上記のほか「当社」または「当社商品」以外の原因（天災等の不可抗力を含む）

5. 責任の制限

本ご承諾事項に記載の保証が、「当社商品」に関する保証のすべてです。

「当社商品」に関連して生じた損害について、「当社」および「当社商品」の販売店は責任を負いません。

6. 輸出管理

「当社商品」または技術資料を、輸出または非居住者に提供する場合は、安全保障貿易管理に関する日本および関係各国の法令・規制を遵守ください。お客様が法令・規則に違反する場合には、「当社商品」または技術資料をご提供できない場合があります。

7. 特許

当社商品の電子および光学に関連する仕様および性能基準は、以下の米国特許により保護されています。

－ 米国特許番号：6,665,621、6,753,776、6,493,653、6,587,811; 7,965,384

他の特許については出願中です。

安全上のご注意

安全に使用していただくための表示と意味について

この取扱説明書では、OS32C を安全にご使用いただくために、注意事項を次のような表示と記号で示しています。ここで示した注意事項は安全に関する重大な内容を記載していますので、必ず守ってください。表示 と記号は次のとおりです。

 警告	正しい取扱いをしなければ、発生しうる危険のために、軽傷・中程度の傷害を負ったり、万一の場合には重傷や死亡にいたる恐れがあります。また、同様に重大な物的損害を受ける恐れがあります。
 注意	正しい取扱いをしなければ、この危険のために、時に軽傷・中程度の傷害を負ったり、あるいは物的損害を受ける恐れがあります。

図記号の意味

	禁止 一般的な禁止を示します。
	特定しない一般的な使用者の行為を指示する図記号

警告表示

 警告	EtherNet/IPを介したシステムおよびエリアの状態に関するモニタ情報は、診断の目的にのみ使用するためのものです。安全に関する機能では使用しないでください。	
	EtherNet/IPを介してモニタされる計測データは、診断の目的にのみ使用するためのものです。安全に関する機能では使用しないでください。	
 注意	計測レポートの設定が、意図した計測データ形式となっているか必ず確認してください。	



目次

1. はじめに	1
2. 検出距離データの精度	2
3. レーザスキャナの設定	3
4. EtherNet/IP 入力アセンブリデータ	4
4.1 表 1 : EtherNet/IP データ型	5
4.2 表 2 : 入力アセンブリ 100 とベンダーオブジェクト 112 (32 バイト)、およびシステム状態	6
4.3 表 3 : 入力アセンブリ 101 (296 バイト)、システムおよび検知状態	8
4.4 表 4 : 出力アセンブリ 113 およびベンダーオブジェクト 115 (104 バイト)、 入力アセンブリ 102、103 用計測レポート設定	10
4.5 表 5: 出力アセンブリ 114 (108 バイト)、入力アセンブリ 104、105 用計測レポート設定	12
4.6 表 6: 出力アセンブリ 115 (316 バイト)、入力アセンブリ 106 ~ 111 用計測レポート設定	14
4.7 表 7 : 出力アセンブリ 112、116 ~ 120 I/O 接続トリガー	17
4.8 表 8 : 共通計測レポートヘッダフォーマット (56 バイト)	20
4.9 表 9 : 入力アセンブリ 102 およびベンダー固有オブジェクト 114 (最大 1410 バイト)	22
4.10 表 10 : 入力アセンブリ 103 およびベンダー固有オブジェクト 116 (最大 1410 バイト)	22
4.11 表 11 : ベンダー固有オブジェクト 117 (最大 2764 バイト)	22
4.12 表 12: 入力アセンブリ 104 (最大 960 バイト)	23
4.13 表 13: 入力アセンブリ 105 (最大 960 バイト)	23
4.14 表 14: 入力アセンブリ 106 (最大 554 バイト)	23
4.15 表 15: 入力アセンブリ 107 (最大 554 バイト)	23
4.16 表 16: 入力アセンブリ 108 (最大 454 バイト)	24
4.17 表 17: 入力アセンブリ 109 (最大 454 バイト)	24
4.18 表 18: 入力アセンブリ 110 (最大 358 バイト)	24
4.19 表 19: 入力アセンブリ 111 (最大 358 バイト)	24
4.20 データリフレッシュ周期 (予測パケット周期)	25
4.21 有効警告エリアセットの選択	25
5. OS32C EDS ファイルのインストール	26
6. CJ シリーズとの通信設定	27
6.1 EtherNet/IP ネットワークをセットアップする	27
6.2 CJ2 の EtherNet/IP タグを設定する	28
6.3 CJ2 に EtherNet/IP 設定をダウンロードする	34
7. NJ シリーズとの通信設定	36
7.1 NJ5 EtherNet/IP アダプタ設定	36
7.2 NJ にタグをセットアップする	37
7.3 タグをグローバル変数に設定する	38
7.4 Network Configurator にタグをエクスポートする	39
7.5 EtherNet/IP ネットワークを設定する	39
7.6 EtherNet/IP 設定を NJ にダウンロードする	42
8. 複数台の OS32C、PLC のセットアップ	44
8.1 1 台の PLC から複数の OS32C をポーリングする	44
8.2 複数の PLC から 1 台の OS32C をポーリングする	49
9. パソコンベースのデバイスとの通信を設定する	51
9.1 EtherNet/IP コマンドプロトコル	51
9.1.1 表 10 : EtherNet/IP データグラムヘッダのコマンドフォーマット	51

9.2 EtherNet/IP コマンドリスト	51
9.2.1 表 11 : EtherNet/IP コマンドリスト	51
9.2.2 表 12 : EtherNet/IP 状態エラーコードリスト	52
9.3 EtherNet/IP コマンド固有データ	52
9.3.1 表 13 : EtherNet/IP 共通パケットフォーマット (CPF)	52
9.4 EtherNet/IP コマンド	53
9.4.1 List Identity コマンド	53
9.4.2 表 14 : List Identity コマンド要求	53
9.4.3 表 15 : List Identity コマンド応答	53
9.4.4 表 16 : EtherNet/IP 識別オブジェクトパラメータ	54
9.4.5 Register Session コマンド	54
9.4.6 表 17 : Register Session コマンド要求	54
9.4.7 表 18 : Register Session コマンド応答	55
9.4.8 Un-Register Session コマンド	55
9.4.9 表 19 : Un-Register Session コマンド要求	56
9.4.10 SendRRData コマンド	56
9.4.11 表 20 : SendRRData コマンド要求	56
9.4.12 表 21 : Get Single Attribute サービスコード要求 CPF データ	57
9.4.13 表 22 : Get Single Attribute 要求に対する SendRRData コマンド応答	57
9.4.14 表 23 : Set Single Attribute サービスコード要求 CPF データ	58
9.4.15 表 24 : Set Single Attribute 要求に対する SendRRData コマンド応答	58
9.4.16 表 25 : Large Forward Open 要求のカプセル化パケット	59
9.4.17 表 26 : Large Forward Open 要求に対する SendRRData コマンド応答	60
9.4.18 表 27 : UDP I/O 接続パケット	61
9.4.19 表 28 : Forward Close 要求のカプセル化パケット	62
9.4.20 表 29 : Forward Close 要求に対する SendRRData コマンド応答	63
10. アプリケーション例	64
10.1 Explicit TCP/IP 要求・返信メッセージを使ったランタイムモニタリング	64
10.1.1 ネットワーク設定	64
10.1.2 パソコン・PLC の設定とプロセス制御の例	64
10.2 Implicit UDP I/O 接続を使ったランタイムモニタリング	72
10.2.1 ネットワーク設定	72
10.2.2 パソコン・PLC の設定とプロセス制御の例	72
11. マニュアル改訂履歴	77

1. はじめに

EtherNet/IP および計測データ対応のセーフティレーザスキャナ形 OS32C-□□□-DM は、EtherNet/IP 通信 ODVA ガイドラインに準拠した製品からのモニタリングが可能です。EtherNet/IP 対応 OS32C が EtherNet/IP オリジネータ（マスタ）デバイスのターゲット（スレーブ）デバイスとして機能します。これにより、複数の EtherNet/IP マスタ デバイスで 1 台の OS32C ユニットのモニタしたり、あるいは、1 台の EtherNet/IP マスタ デバイスで複数の OS32C をモニタすることができます。

セーフティレーザスキャナ形 OS32C-□□□-DM は、システムの状態と検出距離測定データを監視するための EtherNet/IP 通信インターフェースを提供します。EtherNet/IP は、産業環境での使用に適した通信システムです。EtherNet/IP を介して産業用機器は時間のタイムクリティカルなアプリケーション情報を交換できます。この産業用機器には、センサ/アクチュエータなどの単純な入出力装置のほか、ロボット、プログラマブルロジックコントローラ、溶接機、プロセスコントローラなどの複雑な制御機器が含まれます。EtherNet/IP は共通のネットワーク層、トランスポート層、およびアプリケーション層に、ControlNet と DeviceNet においても使用される CIP (Common Industrial Protocol) を使用しています。EtherNet/IP は、標準イーサネットと TCP/IP 技術を活用して CIP 通信パケットを伝送します。これにより、広く普及しているオープンなイーサネットおよび TCP/IP プロトコルの上に、一般的なオープンアプリケーション層が構築されます。EtherNet/IP は、タイムクリティカルな制御データを交換するためのプロデューサ/コンシューマモデルを提供します。プロデューサ/コンシューマモデルを使用すると、データを複数の宛先に複数回送信する必要なく、1 台の送信側デバイス（プロデューサ）と複数台の受信側デバイス（コンシューマ）の間でアプリケーション情報を交換できます。EtherNet/IP の場合、これは、IP マルチキャスト技術とともに CIP ネットワークおよびトランスポート層を利用することで実現されます。単一の生産デバイスが生成した同じアプリケーション情報を複数の EtherNet/IP デバイスが受信できます。EtherNet/IP は標準 IEEE 802.3 を利用しています。時間確定性を改善しようとする非標準の追加はありません。むしろ、EtherNet/IP は、100 Mbps の帯域幅と全二重動作を備えた商用スイッチ技術の使用を推奨することでより時間確定的なパフォーマンスを提供します。

EtherNet/IP 対応 OS32C では、標準 EtherNet/IP による通信を使用し、CIP Safety プロトコルは使用していません。レーザスキャナからの通信データを機能安全目的に使用しないでください。レーザスキャナとの EtherNet/IP 通信は、非安全用途のモニタリングおよび非安全用途である警告エリアの変更専用です。EtherNet/IP のオリジネータデバイスから、OS32C の検知エリアの状態、入出力の状態、設定のチェックサム値などをモニタリングすることが可能です。本書では、OS32C のデータアセンブリに関する詳細、および汎用 PLC と OS32C 間の通信の設定手順などを記載しています。また、OS32C とパソコンベースのデバイス間の通信設定に関する情報なども盛り込んでいます。

2. 検出距離データの精度

検出距離 (mm)		反射体の推奨最小幅 (mm) *1		検出物体の反射率に対する確率1σの推定ランダム誤差 (mm) (A) *2						
				1.8%	5%	10%	25%	50%	75%	100%
250	70	20	10	10	10	10	10	10	10	
500	70	20	10	10	10	10	10	10	10	
1000	70	20	10	10	10	10	10	10	10	
2000	70	20	20	10	10	10	10	10	10	
3000	70	20	20	10	10	10	10	10	10	
4000	70	20	20	20	20	20	20	20	20	
5000	80	30	20	20	20	20	20	20	20	
7500	120	推奨検出距離の範囲外 *3	40	20	20	20	20	20	20	
10000	150		40	20	20	20	20	20	20	
20000	300			40	20	20	20	20	20	
30000	450						30	30	30	
40000	600							30	30	30
50000	750									

*1. 反射体の高さは、アプリケーションや検出距離に応じて300mm以上、1000mm以下としてください。

*2. Aを最大予測誤差とする場合、式 $n \times A + B$ を使用してください。このとき、nはsの任意の乗数（1, 2, 3...）とし、Bは30mmの最大系統誤差とします。検出物体の背景に反射体がある場合は、追加の誤差についてOS32Cユーザーズマニュアルの表7-2を参照してください。光路付近にある物体など、計測条件によっては、これ以外の誤差要素が生じる場合があります。

*3. 最大検出距離は、適用する計測精度や環境条件（煙など）、ウィンドウや対象物の汚れなどによって異なります。

非安全用途の場合、1シグマの値を使用してOS32C-DMの検出距離の精度を求めることができます。この場合、以下の計算式が適用できます。

検出距離の精度 = $(n \times A) + B$

n : 1σ (シグマ)

A : ランダム誤差

B : 系統誤差

たとえば、検出距離1000mmで反射率50%の場合、検出距離の精度は次のように予測できます。

上記の属性値と表から得られる値（A=10 mm、B=30mm）を使用すると、検出距離1000mmで反射率50%の場合精度は、 $(10\text{mm}) + 30\text{mm} = \pm 40\text{mm}$ となります。

3. レーザスキャナの設定

OS32CのIPアドレス設定を変更するだけで、OS32CとEtherNet/IPマスターデバイスの通信が可能です。それ以外の設定変更は不要です。

IPアドレスなど、OS32Cの設定を変更するには、OS32Cユーザズマニュアルの第3章を参照してください。

OS32Cの配線については、OS32Cユーザズマニュアルの第5章を参照してください。

4. EtherNet/IP入力アセンブリデータ

OS32Cでは、下記に示す12の標準入力アセンブリオブジェクトと4つのベンダー固有オブジェクトを提供しています。

- ・入力アセンブリオブジェクト 100 (32バイト) およびベンダー固有オブジェクト 112 : システム状態データを提供します。
- ・入力アセンブリオブジェクト 101 (296バイト) : システム状態およびエリア状態データを提供します。
- ・入力アセンブリオブジェクト 102 (最大1410バイト) およびベンダー固有オブジェクト 114 : システム状態および検出距離計測データを提供します。
- ・入力アセンブリオブジェクト 103 (最大1410バイト) およびベンダー固有オブジェクト 116 : システム状態および反射率計測データを提供します。
- ・入力アセンブリオブジェクト 104 (最大960バイト) : システム状態および検出距離計測データを提供します。
- ・入力アセンブリオブジェクト 105 (最大960バイト) : システム状態および反射率計測データを提供します。
- ・入力アセンブリオブジェクト 106 (最大554バイト) : 検出距離計測データを提供します。
- ・入力アセンブリオブジェクト 107 (最大554バイト) : 反射率計測データを提供します。
- ・入力アセンブリオブジェクト 108 (最大454バイト) : 検出距離計測データを提供します。
- ・入力アセンブリオブジェクト 109 (最大454バイト) : 反射率計測データを提供します。
- ・入力アセンブリオブジェクト 110 (最大358バイト) : 検出距離計測データを提供します。
- ・入力アセンブリオブジェクト 111 (最大358バイト) : 反射率計測データを提供します。
- ・ベンダー固有オブジェクト 117 (最大2764バイト) : システム状態、検出距離計測データ、反射率計測データを提供します。

注 : 「入力オブジェクト」とは、発信側デバイスからみた呼び方です。PLCやクライアントパソコンのソフトウェアアプリケーションは、OS32C側からみた発信側デバイスとなります。

OS32Cが提供する入力アセンブリオブジェクトのデータは、explicit TCP/IP要求・返信メッセージを使うか、あるいは、implicit UDP I/O 接続を使用して一定間隔で繰り返しアセンブリデータを受け取ることができます。OS32Cが提供するベンダー固有オブジェクトのデータは、ソフトウェアアプリケーションで指定した速度による、explicit TCP/IP要求・返信メッセージによって取得することができます。ベンダー固有オブジェクト 114、116および117が提供するデータは、OS32Cのスキャン周期(40ms)と同期しているため、より高い時間精度を要求するアプリケーションでの利用が可能です。リリース3.18以降のファームウェアでは、UDP/IOメッセージも、複数のスキャン周期(40、80、120など)と同期しています。

OS32Cから個々のデータ属性を読み取るには、サービスコード 14 (0x0E) (Get_Single_Attribute) を使ったTCP要求・返信explicitメッセージを使用して、データを取得することができます。

たとえば、以下のパラメータをOS32Cに送信することで、機械の状況が、符号なし16ビットで出力されます。

```
サービスコード 14 (0x0E) // 単一属性取得
オブジェクトクラス 112 (0x70) // ベンダー固有オブジェクト番号、システム状態
インスタンス1 (0x01) // ベンダー固有インスタンス
属性 4 (0x04)、その他の属性については、下記の表を参照。
```

サービスコード 14 (0x0E、Get_Single_Attribute) を使ったTCP要求・返信explicitメッセージを使用して入力アセンブリオブジェクトデータの全容を読み取ることができます。下記のパラメータを使用します。

サービスコード 14 (0x0E) // 単一属性取得
 オブジェクトクラス 4 (0x04) // アセンブリオブジェクトクラス
 インスタンス 100 (0x64) // 入力アセンブリオブジェクト番号、システム状態
 属性 03 (0x03) // 入力アセンブリオブジェクトデータ

ベンダー固有オブジェクトデータの全容を読み取るには、サービスコード 14 (0x0E、Get_Single_Attribute) を使ったTCP要求・返信explicitメッセージを使用することができます。下記のパラメータを使用してください。

サービスコード 14 (0x0E) // 単一属性取得
 オブジェクトクラス 114 (0x72) // ベンダー固有オブジェクト番号（検出距離データ）
 インスタンス 1 (0x01) // ベンダー固有インスタンス
 属性 03 (0x03) // ベンダー固有オブジェクトデータ

4.1 表1：EtherNet/IPデータ型

キーワード	データ型	最小値	最大値
SINT, INT8	Short Integer	-128	127
USINT, UINT8	Unsigned Short Integer	0	255
INT, INT16	Integer	-32768	32767
UINT, UINT16	Unsigned Integer	0	65535
DINT, INT32	Double Integer	-2147483647 (-2^{31})	2147483646 ($2^{31} - 1$)
UDINT, UINT32	Unsigned Double Integer	0	4294967295 (2^{32})
BYTE	Bit string 8-bits	0	0xFF
WORD	Bit string 16-bits	0	0xFFFF
DWORD	Bit string 32-bits	0	0xFFFFFFFF

下記の表2の一覧に示した入力アセンブリ100は、OS32Cの動作や現状の設定のモニタリングに使用できるシステムのランタイム状態を提供します。implicit UDP I/O接続を使用して、アプリケーションで指定した繰返し率によるOS32Cのモニタリングが可能です。

4.2 表2: 入力アセンブリ100とベンダーオブジェクト112(32バイト)、およびシステム状態

ワード# (16ビット)	説明	ベンダー固有オブジェクト0x70、インスタンス0x01属性番号	列举リスト / 可能な値
0	機械の状況 データ型: UINT 16ビット	4	POST = 0 //電源ON自己診断 STOP = 1 //機械停止、防護エリア侵入 INTERLOCK = 2 //スタートインターロック RUN = 3 //機械運転 STANDBY = 4 //システムスタンバイ CONFIGURE = 6 //設定中 FAULT = 7 //システム障害、制御出力OFF
1	機械停止理由 データ型: UINT 16ビット	5	DIRTY_WINDOW = 0 //ウィンドウ汚れによるシステム停止 ZONE_SELECT_NUM_ACTIVE = 1 //有効入力数間違いによる無効なエリアセット選択入力 ZONE_SELECT_INVALID = 2 //正しい有効入力数による無効なエリアセット選択入力 QUALIFIED_ZONE_VIOLATION = 3 //条件付き防護エリア侵入 STAND_BY_REQUEST = 4 //スタンバイの要求 RX_ERROR = 5 //受信側エラー TOO_CLEAN_WINDOW = 6 //清潔過ぎるウィンドウによるシステム停止 NOT_IN_MACHINE_STOP_1 = 7 //システムが現在機械停止状態にない SYSTEM_FAULTED = 8 //システム障害、診断コードを確認 NOT_IN_MACHINE_STOP_2 = 9 //システムが現在機械停止状態にない
2	有効防護エリアセット データ型: UINT 16ビット	6	AZS_01 = 0 (0x00) ... AZS_70 = 69 (0x45) INVALID AZS = 32,768 (0x8000)
3	エリアセット選択入力 データ型: WORD 16ビット	7	エリアセット選択入力1 = ビット0 エリアセット選択入力2 = ビット1 エリアセット選択入力3 = ビット2 エリアセット選択入力4 = ビット3 エリアセット選択入力5 = ビット4 エリアセット選択入力6 = ビット5 エリアセット選択入力7 = ビット6 エリアセット選択入力8 = ビット7
4	検知エリア状態 データ型: WORD 16ビット	8	防護エリア = ビット0 警告エリア1 = ビット1 警告エリア2 = ビット2 ウィンドウ汚れ = ビット3
5	出力状態 データ型: WORD 16ビット	9	制御出力 = ビット0 補助出力 = ビット1 警告出力 = ビット2
6	入力状態 データ型: WORD 16ビット	10 (0x0A)	スタンバイ入力 = ビット0 スタート入力 = ビット1 EDM入力 = ビット2
7	7セグメント表示 データ型: UINT 16ビット	11 (0x0B)	下位桁から上位桁へ 各桁の通常表示は、1～9 (0x01～0x09)。機械停止中は、0x1B 0x1Bでダッシュ(--)を表示。

ワード# (16ビット)	説明	ベンダー固有オブジェクト0x70、インスタンス0x01属性番号	列挙リスト / 可能な値
8	非安全設定 チェックサム値 データ型: UINT 16ビット	12 (0x0C)	16ビット 非安全設定CRC値
9	安全設定 チェックサム値 データ型: UINT 16ビット	13 (0x0D)	16ビット 安全設定CRC値
10	有効警告エリア セット データ型: UNIT 16ビット	14 (0x0E)	AZS_01 = 0 (0x00) ... AZS_70 = 69 (0x45) INVALID AZS = 32,768 (0x8000)

注意: ワード番号11~15は使用しません。

下記の表3の一覧に示した入力アセンブリ101は、システムのランタイム状態に加えて、防護エリア、警告エリア1、警告エリア2、検知状態を提供します。これらの情報を使って、OS32Cの動作や検知エリアをモニタリングすることができます。また、implicit UDP I/O接続を使用して、アプリケーションで指定した繰返し率によるOS32Cのモニタリングが可能です。

4.3 表3：入力アセンブリ101（296 バイト）、システムおよび検知状態

ワード# (16ビット)	属性説明	データフィールド (16ビット)
0	機械の状況	POST = 0 //電源ON自己診断 STOP = 1 //機械停止、防護エリア侵入 INTERLOCK = 2 //スタートインターロック RUN = 3 //機械運転 STANDBY = 4 //システムスタンバイ CONFIGURE = 6 //設定中 FAULT = 7 //システム障害、制御出力OFF
1	機械停止理由	DIRTY_WINDOW = 0 //ウィンドウ汚れによるシステム停止 ZONE_SELECT_NUM_ACTIVE = 1 //有効入力数間違いによる無効なエリアセット選択入力 ZONE_SELECT_INVALID = 2 //正しい有効入力数による無効なエリアセット選択入力 QUALIFIED_ZONE_VIOLATION = 3 //条件付き防護エリア侵入 STAND_BY_REQUEST = 4 //スタンバイの要求 RX_ERROR = 5 //受信側エラー TOO_CLEAN_WINDOW = 6 //清潔過ぎるウィンドウによるシステム停止 NOT_IN_MACHINE_STOP_1 = 7 //システムが現在機械停止状態にない SYSTEM_FAULTED = 8 //システム障害、診断コードを確認 NOT_IN_MACHINE_STOP_2 = 9 //システムが現在機械停止状態にない
2	有効防護エリアセット	AZS_01 = 0 (0x00) ... AZS_70 = 69 (0x45) INVALID AZS = 32,768 (0x8000)
3	エリアセット選択入力	エリアセット選択1 = ビット0 エリアセット選択2 = ビット1 エリアセット選択3 = ビット2 エリアセット選択4 = ビット3 エリアセット選択5 = ビット4 エリアセット選択6 = ビット5 エリアセット選択7 = ビット6 エリアセット選択8 = ビット7
4	検知エリア状態	防護エリア = ビット0 警告エリア1 = ビット1 警告エリア2 = ビット2 ウィンドウ汚れ = ビット3
5	出力状態	制御出力 = ビット0 補助出力 = ビット1 警告出力 = ビット2
6	入力状態	スタンバイ入力 = ビット0 スタート入力 = ビット1 EDM入力 = ビット2
7	7セグメント表示	下位桁から上位桁へ 各桁の通常表示は、1～9 (0x01～0x09)。機械停止中は、0x1B 0x1Bでダッシュ (—)を表示。
8	非安全設定 チェックサム値	16ビット 非安全設定CRC値
9	安全設定 チェックサム値	16ビット 安全設定CRC値

ワード# (16ビット)	属性説明	データフィールド (16ビット)
10	有効警告エリアセット データ型: UNIT 16ビット	AZS_01 = 0 (0x00) ... AZS_70 = 69 (0x45) INVALID AZS = 32,768 (0x8000)
11	未使用	将来的に使用
12	未使用	将来的に使用
13	未使用	将来的に使用
14	未使用	将来的に使用
15	未使用	将来的に使用
16	防護エリア状態 ビーム1 (ビーム1～16)	ビーム状態 ビット0～ビット16 (0 = 遮断なし、1 = 遮断)
17	防護エリア状態 ビーム2 (ビーム17～32)	ビーム状態 ビット0～ビット16 (0 = 遮断なし、1 = 遮断)
18～57	防護エリア状態 ビーム3 (ビーム33～48) から 防護エリア状態 ビーム42 (ビーム653～672)	ビーム状態 ビット0～ビット16 (0 = 遮断なし、1 = 遮断)
58	防護エリア状態 ビーム43 (ビーム673～688)	ビーム状態 ビット0～ビット16 (0 = 遮断なし、1 = 遮断) ビーム678～688は未使用 未使用ビームは0に設定 注意：データは16ビットワード整列の維持に使用
59	防護エリア状態 ビーム44 (ビーム689～704)	未使用ビームは0に設定 注意：データは32ビット整列の維持に使用
60	警告エリア1状態 ビーム1 (ビーム1～16)	ビーム状態 ビット0～ビット16 (0 = 遮断なし、1 = 遮断)
61	警告エリア1状態 ビーム2 (ビーム17～32)	ビーム状態 ビット0～ビット16 (0 = 遮断なし、1 = 遮断)
62～101	警告エリア1状態 ビーム3 (ビーム33～48) から 警告エリア1状態 ビーム42 (ビーム653～672)	ビーム状態 ビット0～ビット16 (0 = 遮断なし、1 = 遮断)
102	警告エリア1状態 ビーム43 (ビーム672～688)	ビーム状態 ビット0～ビット16 (0 = 遮断なし、1 = 遮断) ビーム678～688は未使用 未使用ビームは0に設定 注意：データは16ビットワード整列の維持に使用
103	警告エリア1状態 ビーム44 (ビーム689～704)	未使用ビームは0に設定 注意：データは32ビット整列の維持に使用
104	警告エリア2状態 ビーム1 (ビーム1～16)	ビーム状態 ビット0～ビット16 (0 = 遮断なし、1 = 遮断)
105	警告エリア2状態 ビーム2 (ビーム17～32)	ビーム状態 ビット0～ビット16 (0 = 遮断なし、1 = 遮断)
106～145	警告エリア2状態 ビーム3 (ビーム33～48) から 警告エリア2状態 ビーム42 (ビーム653～672)	ビーム状態 ビット0～ビット16 (0 = 遮断なし、1 = 遮断)
146	警告エリア2状態 ビーム43 (ビーム672～688)	ビーム状態 ビット0～ビット16 (0 = 遮断なし、1 = 遮断) ビーム678～688は未使用 未使用ビームは0に設定 注意：データは16ビットワード整列の維持に使用
147	警告エリア2状態 ビーム44 (ビーム689～704)	未使用ビームは0に設定 注意：データは32ビット整列の維持に使用

下記の表4の一覧に示した出力アセンブリオブジェクト113とベンダー固有オブジェクト115は、入力アセンブリオブジェクト102、103およびベンダー固有オブジェクト114、116、117においてOS32Cが提供するランタイム計測データの設定に使用します。入力アセンブリ113とベンダー固有オブジェクト115においてビームレポート選択マスク配列を使用することで、計測レポートの検出距離フォーマットおよび反射率フォーマット、およびモニタリングする個々のビームを選択することができます。

4.4 表4：出力アセンブリ113およびベンダーオブジェクト115（104バイト）、入力アセンブリ102、103用計測レポート設定

ワード# (16ビット)	説明	ベンダー固有オブジェクト x、インスタンス y、属性番号 z	列挙リスト / 可能な値
0	検出距離レポート フォーマット データ型：UINT 16ビット	0x73, 1, 4	NO_TOF_MEASUREMENTS = 0, //飛行時間（TOF）計測不要 RANGE_MEASURE_50M = 1, //デフォルト設定 //ビット0～15（16ビット）距離計測値（0～50,000mm） //値：0x0001 = ノイズビーム、値：0xFFFF = 反射なし RANGE_MEASURE_32M_PZ = 2, //ビット0～14（15ビット）距離計測値（0～32,766mm） //値：0x0001 = ノイズビーム、値：0x7FFF = 反射なし //ビット15：防護エリア内での物体検知（ビーム遮断） RANGE_MEASURE_16M_WZ1PZ = 3, //ビット0～13（14ビット）距離計測値（0～16,382mm） //値：0x0001 = ノイズビーム、値：0x3FFF = 反射なし //ビット14：警告エリア1内での物体検知（ビーム遮断） //ビット15：防護エリア内での物体検知（ビーム遮断） RANGE_MEASURE_8M_WZ2WZ1PZ = 4, //ビット0～12（13ビット）距離計測値（0～8,190mm） //値：0x0001 = ノイズビーム、値：0x1FFF = 反射なし //ビット13：警告エリア2内での物体検知（ビーム遮断） //ビット14：警告エリア1内での物体検知（ビーム遮断） //ビット15：防護エリア内での物体検知（ビーム遮断） RANGE_MEASURE_TOF_4PS = 5 //ビット0～15（16ビット）TOF計測値（0～65,534 x 4ps） //値：0x0001 = ノイズビーム、値：0xFFFF = 反射なし RANGE_MEASURE_50M_W3BIT_ENCODED_TOT = 6, //ビット0～12（13ビット）距離計測値 //（0～50,000mm、分解能8mmで50m） //ビット13～15（3ビット）コード化された反射率値（計測値の1/128） RANGE_MEASURE_32M_WZ2 = 7, //ビット0～14（15ビット）距離計測値（0～32,766mm） //ビット15：警告エリア2内での計測値検知（ビーム遮断） RANGE_MEASURE_16M_WZ1WZ2 = 8, //ビット0～13（14ビット）距離計測値（0～16,382mm） //ビット14：警告エリア1内での計測値検知（ビーム遮断） //ビット15：警告エリア2内での計測値検知（ビーム遮断） RANGE_MEASURE_8M_PZWZ1WZ2 = 9, //ビット0～12（13ビット）距離計測値（0～8,190mm） //ビット13：防護エリア内での計測値検知（ビーム遮断） //ビット14：警告エリア1内での計測値検知（ビーム遮断） //ビット15：警告エリア2内での計測値検知（ビーム遮断）

ワード# (16ビット)	説明	ベンダー固有オブ ジェクト x、インスタ ンス y、属性番号 z	列挙リスト / 可能な値
1	反射率レポート フォーマット データ型: UINT 16ビット	0x73, 1, 5	NO_TOT_MEASUREMENTS = 0, //しきい値超過時間 (TOT) 計測不要 REFLECTIVITY_MEASURE_TOT_ENCODED = 1, //デフォルト設定 //ビット0~9 (10ビット) TOTスケール値 (0~1,000) //ビット10: 未使用 //ビット11: 警告エリア2内での物体検知 (ビーム遮断) //ビット12: 警告エリア1内での物体検知 (ビーム遮断) //ビット13: 防護エリア内での物体検知 (ビーム遮断) //ビット14: ノイズビーム検知 //ビット15: 反射検知なし REFLECTIVITY_MEASURE_TOT_4PS = 2 //ビット0~15 (16ビット) TOT計測値 (0~65,535 x 4ps)
2	検出距離レポート モード データ型: UNIT 16 ビット	0x73, 1, 6	REPORT_RANGE_ONLY = 0 (default), // アセンブリ102 検出距離レポートのみ REPORT_RANGE_AND_REFLECTIVITY = 1, // アセンブリ102 検出距離レポートと反射率レポート
3	有効警告エリア セット番号 データ型: UNIT 16 ビット	0x73, 1, 7	最小値: ZONE_1 = 0, ... 最大値: ZONE_70 = 69.
4	未使用		将来的に使用
5	未使用		将来的に使用
6	未使用		将来的に使用
7	未使用		将来的に使用
8 - 51	ビームレポート選 択マスク データ型: UINT 16ビット ARRAY[44]	0x73, 1, 12 (0x0C)	ビームレポート選択マスクは、アセンブリ入力オブジェクト102、103およ びベンダー固有オブジェクト114、116、117にてレポートするビームの計測 値の定義に使用。 ビット = 0、計測レポートより除外 //デフォルト設定 ビット = 1、計測レポートに含む

4.5 表5: 出力アセンブリ114（108バイト）、入力アセンブリ104、105 用計測レポート設定

ワード# (16ビット)	説明	列挙リスト / 可能な値
0	検出距離レポート フォーマット データ型: UINT 16ビット	NO_TOF_MEASUREMENTS = 0, // 飛行時間 (TOF) 計測不要 RANGE_MEASURE_50M = 1, // デフォルト設定 // ビット0～15 (16ビット) 距離計測値 (0～50,000mm) // 値: 0x0001 = ノイズビーム、値: 0xFFFF = 反射なし RANGE_MEASURE_32M_PZ = 2, // ビット0～14 (15ビット) 距離計測値 (0～32,766mm) // 値: 0x0001 = ノイズビーム、値: 0x7FFF = 反射なし // ビット15: 防護エリア内での物体検知 (ビーム遮断) RANGE_MEASURE_16M_WZ1PZ = 3, // ビット0～13 (14ビット) 距離計測値 (0～16,382mm) // 値: 0x0001 = ノイズビーム、値: 0x3FFF = 反射なし // ビット14: 警告エリア1内での物体検知 (ビーム遮断) // ビット15: 防護エリア内での物体検知 (ビーム遮断) RANGE_MEASURE_8M_WZ2WZ1PZ = 4, // ビット0～12 (13ビット) 距離計測値 (0～8,190mm) // 値: 0x0001 = ノイズビーム、値: 0x1FFF = 反射なし // ビット13: 警告エリア2内での物体検知 (ビーム遮断) // ビット14: 警告エリア1内での物体検知 (ビーム遮断) // ビット15: 防護エリア内での物体検知 (ビーム遮断) RANGE_MEASURE_TOF_4PS = 5 // ビット0～15 (16ビット) TOF計測値 (0～65,534 x 4ps) // 値: 0x0001 = ノイズビーム、値: 0xFFFF = 反射なし RANGE_MEASURE_50M_W3BIT_ENCODED_TOT = 6, // ビット0～12 (13ビット) 距離計測値 // (0～50,000mm、分解能8mmで50m) // ビット13～15 (3ビット) コード化された反射率値 (計測値の1/128) RANGE_MEASURE_32M_WZ2 = 7, // ビット0～14 (15ビット) 距離計測値 (0～32,766mm) // ビット15: 警告エリア2内での計測値検知 (ビーム遮断) RANGE_MEASURE_16M_WZ1WZ2 = 8, // ビット0～13 (14ビット) 距離計測値 (0～16,382mm) // ビット14: 警告エリア1内での計測値検知 (ビーム遮断) // ビット15: 警告エリア2内での計測値検知 (ビーム遮断) RANGE_MEASURE_8M_PZWZ1WZ2 = 9, // ビット0～12 (13ビット) 距離計測値 (0～8,190mm) // ビット13: 防護エリア内での計測値検知 (ビーム遮断) // ビット14: 警告エリア1内での計測値検知 (ビーム遮断) // ビット15: 警告エリア2内での計測値検知 (ビーム遮断)

ワード# (16ビット)	説明	列挙リスト / 可能な値
1	反射率レポート フォーマット データ型: UINT 16ビット	NO_TOT_MEASUREMENTS = 0, // しきい値超過時間 (TOT) 計測不要 REFLECTIVITY_MEASURE_TOT_ENCODED = 1, // デフォルト設定 // ビット0~9 (10ビット) TOTスケール値 (0~1,000) // ビット10: 未使用 // ビット11: 警告エリア2内での物体検知 (ビーム遮断) // ビット12: 警告エリア1内での物体検知 (ビーム遮断) // ビット13: 防護エリア内での物体検知 (ビーム遮断) // ビット14: ノイズビーム検知 // ビット15: 反射検知なし REFLECTIVITY_MEASURE_TOT_4PS = 2 // ビット0~15 (16ビット) TOT計測値 (0~65,535 x 4ps)
2	検出距離 レポートモード データ型: UINT 16ビット	REPORT_RANGE_ONLY = 0, // 検出距離レポートのみ REPORT_RANGE_AND_REFLECTIVITY = 1, // 検出距離レポートと反射率レポート
3	アセンブリ104、105 ユーザータグ有効 データ型: UINT 16ビット	入力アセンブリデータの先頭位置および最終位置を示すためのユーザ設定のタグ値を有効化 // 無効 = 0 (デフォルト)、有効 = 1
4	アセンブリ104 ユーザーデータの先頭 位置情報 データ型: UINT 16ビット	入力アセンブリデータの先頭位置を示すためのユーザ設定のタグ値
5	アセンブリ104 ユーザーデータの最終 位置情報 データ型: UINT 16ビット	入力アセンブリデータの最終位置を示すためのユーザ設定のタグ値
6	アセンブリ105 ユーザーデータの先頭 位置情報 データ型: UINT 16ビット	入力アセンブリデータの先頭位置を示すためのユーザ設定のタグ値
7	アセンブリ105 ユーザーデータの最終 位置情報 データ型: UINT 16ビット	入力アセンブリデータの最終位置を示すためのユーザ設定のタグ値
8	有効警告エリアセッ ト番号 データ型: UINT 16ビット	最小値: ZONE_1 = 0, ... 最大値: ZONE_70 = 69.
9	未使用	将来的に使用
10- 53	ビームレポート選択 マスク データ型: UINT 16ビット ARRAY[44]	ビームレポート選択マスクは、アセンブリ入力オブジェクト104、105にてレポートするビームの 計測値の定義に使用。 ビット = 0、計測レポートから除外 ビット = 1、計測レポートに含む// デフォルト設定

4.6 表6: 出力アセンブリ115（316バイト）、入力アセンブリ106～111 用計測レポート設定

ワード# (16ビット)	説明	列挙リスト / 可能な値
0	検出距離レポート フォーマット データ型: UINT 16ビット	NO_TOF_MEASUREMENTS = 0, // 飛行時間 (TOF) 計測不要 RANGE_MEASURE_50M = 1, // デフォルト設定 // ビット0～15 (16ビット) 距離計測値 (0～50,000mm) // 値: 0x0001 = ノイズビーム、値: 0xFFFF = 反射なし RANGE_MEASURE_32M_PZ = 2, // ビット0～14 (15ビット) 距離計測値 (0～32,766mm) // 値: 0x0001 = ノイズビーム、値: 0x7FFF = 反射なし // ビット15: 防護エリア内での物体検知 (ビーム遮断) RANGE_MEASURE_16M_WZ1PZ = 3, // ビット0～13 (14ビット) 距離計測値 (0 to 16,382 millimeters). // 値: 0x0001 = ノイズビーム、値: 0x3FFF = 反射なし // ビット14: 警告エリア1内での物体検知 (ビーム遮断) // ビット15: 防護エリア内での物体検知 (ビーム遮断) RANGE_MEASURE_8M_WZ2WZ1PZ = 4, // ビット0～12 (13ビット) 距離計測値 (0～8,190mm) // 値: 0x0001 = ノイズビーム、値: 0x1FFF = 反射なし // ビット13: 警告エリア2内での物体検知 (ビーム遮断) // ビット14: 警告エリア1内での物体検知 (ビーム遮断) // ビット15: 防護エリア内での物体検知 (ビーム遮断) RANGE_MEASURE_TOF_4PS = 5 // ビット0～15 (16ビット) TOF計測値 (0～65,534 x 4ps) // 値: 0x0001 = ノイズビーム、値: 0xFFFF = 反射なし RANGE_MEASURE_50M_W3BIT_ENCODED_TOT = 6, // ビット0～12 (13ビット) 距離計測値 // (0～50,000mm、分解能8mmで50m) // ビット13～15 (3ビット) コード化された反射率値 (計測値の1/128) RANGE_MEASURE_32M_WZ2 = 7, // ビット0～14 (15ビット) 距離計測値 (0～32,766mm) // ビット15: 警告エリア2内での計測値検知 (ビーム遮断) RANGE_MEASURE_16M_WZ1WZ2 = 8, // ビット0～13 (14ビット) 距離計測値 (0～16,382mm) // ビット14: 警告エリア1内での計測値検知 (ビーム遮断) // ビット15: 警告エリア2内での計測値検知 (ビーム遮断) RANGE_MEASURE_8M_PZWZ1WZ2 = 9, // ビット0～12 (13ビット) 距離計測値 (0 to 8,190 millimeters). // ビット13: 防護エリア内での計測値検知 (ビーム遮断) // ビット14: 警告エリア1内での計測値検知 (ビーム遮断) // ビット15: 警告エリア2内での計測値検知 (ビーム遮断)

ワード# (16ビット)	説明	列挙リスト / 可能な値
1	反射率レポート フォーマット データ型: UINT 16ビット	NO_TOT_MEASUREMENTS = 0, // しきい値超過時間 (TOT) 計測不要 REFLECTIVITY_MEASURE_TOT_ENCODED = 1, // デフォルト設定 // ビット0~9 (10ビット) TOTスケール値 (0~1,000) // ビット10: 未使用 // ビット11: 警告エリア2内での物体検知 (ビーム遮断) // ビット12: 警告エリア1内での物体検知 (ビーム遮断) // ビット13: 防護エリア内での物体検知 (ビーム遮断) // ビット14: ノイズビーム検知 // ビット15: 反射検知なし REFLECTIVITY_MEASURE_TOT_4PS = 2 // ビット0~15 (16ビット) TOT計測値 (0~65,535 x 4ps)
2	アセンブリ106 検出距離レポート モード データ型: UINT 16ビット	REPORT_RANGE_ONLY = 0, // 検出距離レポートのみ REPORT_RANGE_AND_REFLECTIVITY = 1, // 検出距離レポートと反射率レポート
3	アセンブリ106、107 ユーザータグ有効 データ型: UINT 16ビット	入力アセンブリデータの先頭位置および最終位置を示すためのユーザ設定のタグ値を有効化 // 無効 = 0 (デフォルト)、有効 = 1
4	アセンブリ106 ユーザーデータの先頭 位置情報 データ型: UINT 16ビット	入力アセンブリデータの先頭位置を示すためのユーザ設定のタグ値
5	アセンブリ106 ユーザーデータの最終 位置情報 データ型: UINT 16ビット	入力アセンブリデータの最終位置を示すためのユーザ設定のタグ値
6	アセンブリ107 ユーザーデータの先頭 位置情報 データ型: UINT 16ビット	入力アセンブリデータの先頭位置を示すためのユーザ設定のタグ値
7	アセンブリ107 ユーザーデータの最終 位置情報 データ型: UINT 16ビット	入力アセンブリデータの最終位置を示すためのユーザ設定のタグ値
8	有効警告エリアセッ ト番号 データ型: UINT 16ビット	最小値: ZONE_1 = 0, ... 最大値: ZONE_70 = 69.
9	未使用	将来的に使用
10- 53	ビームレポート選択 マスク データ型: UINT 16ビット ARRAY[44]	ビームレポート選択マスクは、アセンブリ入力オブジェクト106、107にてレポートするビーム の計測値の定義に使用。 ビット = 0、計測レポートから除外 ビット = 1、計測レポートに含む// デフォルト設定
54	アセンブリ108 検知距離レポート モード データ型: UINT 16ビット	REPORT_RANGE_ONLY = 0, // 検知距離レポートのみ REPORT_RANGE_AND_REFLECTIVITY = 1, // 検知距離レポートと反射率レポート

ワード# (16ビット)	説明	列挙リスト / 可能な値
55	アセンブリ108、109 ユーザータグ有効 データ型: UINT 16ビット	入力アセンブリデータの先頭位置および最終位置を示すためのユーザ設定のタグ値を有効化 // 無効 = 0 (デフォルト)、有効 = 1
56	アセンブリ108 ユーザーデータの先頭 位置情報 データ型: UINT 16ビット	入力アセンブリデータの先頭位置を示すためのユーザ設定のタグ値
57	アセンブリ108 ユーザーデータの最終 位置情報 データ型: UINT 16ビット	入力アセンブリデータの最終位置を示すためのユーザ設定のタグ値
58	アセンブリ109 ユーザーデータの先頭 位置情報 データ型: UINT 16ビット	入力アセンブリデータの先頭位置を示すためのユーザ設定のタグ値
59	アセンブリ109 ユーザーデータの最終 位置情報 データ型: UINT 16ビット	入力アセンブリデータの最終位置を示すためのユーザ設定のタグ値
60	未使用	将来的に使用
61	未使用	将来的に使用
62-105	ビームレポート選択 マスク データ型: UINT 16ビット ARRAY[44]	ビームレポート選択マスクは、アセンブリ入力オブジェクト108、109にてレポートするビーム の計測値の定義に使用。 ビット = 0、計測レポートから除外 ビット = 1、計測レポートに含む// デフォルト設定
106	アセンブリ110 検出距離レポート モード データ型: UINT 16ビット	REPORT_RANGE_ONLY = 0, // 検出距離レポートのみ REPORT_RANGE_AND_REFLECTIVITY = 1, // 検出距離レポートと反射率レポート
107	アセンブリ110、111 ユーザータグ有効 データ型: UINT 16ビット	入力アセンブリデータの先頭位置および最終位置を示すためのユーザ設定のタグ値を有効化 // 無効 = 0 (デフォルト)、有効 = 1
108	アセンブリ110 ユーザーデータの先頭 位置情報 データ型: UINT 16ビット	入力アセンブリデータの先頭位置を示すためのユーザ設定のタグ値
109	アセンブリ110 ユーザーデータの最終 位置情報 データ型: UINT 16ビット	入力アセンブリデータの最終位置を示すためのユーザ設定のタグ値
110	アセンブリ111 ユーザーデータの先頭 位置情報 データ型: UINT 16ビット	入力アセンブリデータの先頭位置を示すためのユーザ設定のタグ値

ワード# (16ビット)	説明	列挙リスト / 可能な値
111	アセンブリ111 ユーザデータの最終 位置情報 データ型: UINT 16ビット	入力アセンブリデータの最終位置を示すためのユーザ設定のタグ値
112	未使用	将来的に使用
113	未使用	将来的に使用
114-157	ビームレポート選択 マスク データ型: UINT 16ビット ARRAY[44]	ビームレポート選択マスクは、アセンブリ入力オブジェクト110、111にてレポートするビーム の計測値の定義に使用。 ビット = 0、計測レポートから除外 ビット = 1、計測レポートに含む// デフォルト設定

設定出力アセンブリ113、114、115の検出距離レポートフォーマットでは、検出距離計測値をmm単位、コード化された検知エリアビットとmm単位、あるいは、ピコ秒単位でのTOF計測値として受け取ることができます。OS32Cからのコード化された検知エリアビットを使用すると、1) エリア侵入が発生する場所を即座に検知できる、2) クライアント（PLCあるいはパソコン）にOS32Cで保存されている検知エリアパラメータと同一のパラメータを持たせる必要がないというメリットがあります。また、検出距離計測値をコード化することによって、選択したフォーマット用のバイナリマスクを使用して、計測値を検知エリアビットから抜き出すことができます。計測ビームにノイズが多く有効な値が得られない場合、1（0x0001）の値を返し、値の計測が得られない場合は、検出距離レポートフォーマットの最大値が返されます（例：RANGE_MEASURE_50Mフォーマットの場合、65535（0xFFFF）が返される）。検出距離計測値は、入力アセンブリオブジェクト102、104、106、108、110、およびベンダー固有オブジェクト114、117で提供されます。ベンダー固有オブジェクト117では、反射率計測値も提供されています。

設定出力アセンブリ113、114、115の反射率レポートフォーマットでは、反射率計測値をコード化された検知エリアビットとスケール単位のデータ、または、ピコ秒単位でのTOT計測値としても受け取ることができます。反射率計測に組み込まれたコード化された検知ビットには、前述のようなメリットがありますが、それ以外にも、長距離の計測が必要な場合のエリア検知方法としても使用することができます。反射率計測値は、入力アセンブリオブジェクト103、105、107、109、111、およびベンダー固有オブジェクト116、117で提供されます。ベンダー固有オブジェクト117では、検出距離計測値も提供されています。

出力アセンブリ112および116から120までは単一ワードアセンブリです。標準入力アセンブリ(100～111)のストリーミングを行うPLC I/O接続をキープアライブするために使用されます。このアセンブリの出力データは計測レポートの設定に影響しないため、別のI/O接続のトリガーとして使用できます。たとえば、計測レポート設定の出力アセンブリ113を使用して検出距離計測データを収集していると、出力アセンブリ112および116から120までを使用して別のストリーミング接続を確立し、反射率データを収集することができます。

4.7 表7：出力アセンブリ112、116～120 I/O接続トリガー

ワード# (16ビット)	説明	列挙リスト / 可能な値
0	データ型: UINT 16ビット	未使用

OS32Cでは、ビームレポート選択マスクを使って対象領域を選択します。ビームレポート選択マスクを使用することで、ランタイム中のモニタリングエリア変更が可能で、柔軟な分解能の設定や対象領域の選択ができます。下記の図4-1では、すべてのビームを選択し、ビームレポート選択マスクの個々のビットを1に設定した場合の適用領域を示しています。安全領域全体が常時必ず保護状態に置かれるようにするため、検知フィールドの前後には追加ビームを設けています。したがって、実際のモニタリング領域は -0.4° から 270.4° となります。

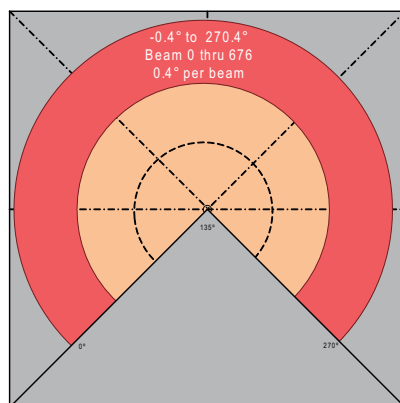


図4-1 ビームレポート選択マスク：全ビーム選択

OS32Cのビームレポート選択マスクを利用して、より狭い対象エリアを設定できます。たとえば、ビームレポート選択マスクの225から450ビットを選択する設定では、走査面の 90° から 180° の領域をカバーすることができます。

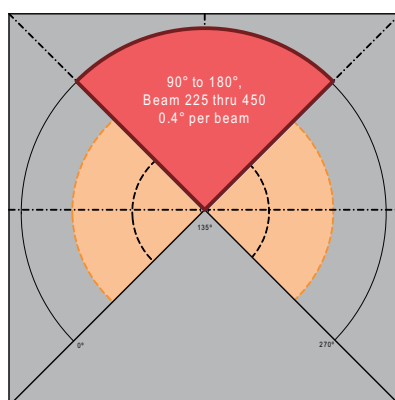


図4-2 ビーム出力選択マスク：ビーム225～450選択

下の図4-3のように、ビームレポート選択マスクを使って、対象領域のデータ量を減らすこともできます。たとえば、ビームレポート選択マスクの225から450ビットのうち、5番目ごとのビームを設定することで、分解能 2° の対象領域を設定することができます。

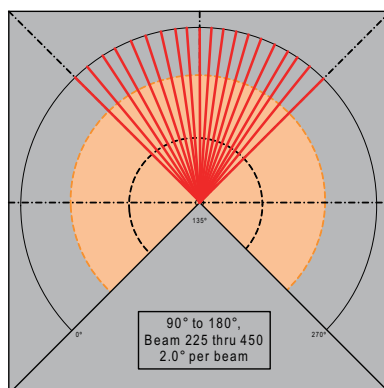


図4-3 ビームレポート選択マスク (低分解能) : ビーム225~450

入力アセンブリオブジェクト102～105およびベンダー固有オブジェクト114、116、117では、それぞれのアセンブリ構造に含まれる固有の計測データに加えて、共通計測レポートヘッダフォーマットが提供されています。下の表8は、共通の計測レポートヘッダフォーマットです。アセンブリオブジェクト102あるいは103に対して、explicit TCP/IP要求・返信メッセージまたはimplicit UDP I/O接続を使用し、アプリケーションで指定した繰返し率によるOS32Cのモニタリングが可能です。

4.8 表8：共通計測レポートヘッダフォーマット（56バイト）

ワード# (16ビット)	説明	ベンダー固有オブジェクト x、インスタンス y、属性番号 z	列挙リスト / 可能な値
0 & 1	スキャン数 データ型：UDINT 32ビット	0x72, 1, 13 (0x0D)	0～4294967295 (2 ³²)
2 & 3	スキャン周期 データ型：UDINT 32ビット	0x72, 1, 14 (0x0E)	39,000 us ± 500 us
4 & 5	スキャン時刻情報 データ型：UDINT 32ビット	0x72, 1, 15 (0x0F)	0～4294967295 (2 ³²) us
6 & 7	スキャンビーム周期 データ型：UDINT 32ビット	0x72, 1, 16 (0x10)	42,777～43,888 ns
8	機械の状況 データ型：UINT 16ビット	0x70, 1, 4	POST = 0 //電源ON自己診断 STOP = 1 //機械停止、防護エリア侵入 INTERLOCK = 2 //スタートインターロック RUN = 3 //機械運転 STANDBY = 4 //システムスタンバイ CONFIGURE = 6 //設定中 FAULT = 7 //システム障害、制御出力OFF
9	機械停止理由 データ型：UINT 16ビット	0x70, 1, 5	DIRTY_WINDOW = 0 //ウィンドウ汚れによる機械停止 ZONE_SELECT_NUM_ACTIVE = 1 //有効入力数間違いによる無効なエリアセット選択入力 ZONE_SELECT_INVALID = 2 //正しい有効入力数による無効なエリアセット選択入力 QUALIFIED_ZONE_VIOLATION = 3 //条件付き防護エリア侵入 STAND_BY_REQUEST = 4 //スタンバイの要求 RX_ERROR = 5 //受信側エラー TOO_CLEAN_WINDOW = 6 //清潔過ぎるウィンドウによる機械停止 NOT IN MACHINE STOP_1 = 7 //システムが現在機械停止状態にない SYSTEM FAULTED = 8 //システム障害、診断コードを確認NOT NOT IN MACHINE STOP_2 = 9 //システムが現在機械停止状態にない
10	有効防護エリアセット データ型：UINT 16ビット	0x70, 1, 6	AZS_01 = 0 (0x00) ... AZS_70 = 69 (0x45) INVALID AZS = 32,768 (0x8000)
11	エリアセット選択入力 データ型：WORD 16ビット	0x70, 1, 7	エリアセット選択1 = ビット0 エリアセット選択2 = ビット1 エリアセット選択3 = ビット2 エリアセット選択4 = ビット3 エリアセット選択5 = ビット4 エリアセット選択6 = ビット5 エリアセット選択7 = ビット6 エリアセット選択8 = ビット7
12	検知エリア状態 データ型：WORD 16ビット	0x70, 1, 8	防護エリア = ビット0 警告エリア1 = ビット1 警告エリア2 = ビット2 ウィンドウ汚れ = ビット3

ワード# (16ビット)	説明	ベンダー固有オブジェクト x、インスタンス y、属性番号 z	列挙リスト / 可能な値
13	出力状態 データ型: WORD 16ビット	0x70, 1, 9	制御出力 = ビット0 補助出力 = ビット1 警告出力 = ビット2
14	入力状態 データ型: WORD 16ビット	0x70, 1, 10 (0x0A)	スタンバイ入力 = ビット0 スタート入力 = ビット1 EDM入力 = ビット2
15	7セグメント表示 データ型: UINT 16ビット	0x70, 1, 11 (0x0B)	下位桁から上位桁へ 各桁の通常表示は、1~9 (0x01~0x09)。 機械停止中は、0x1B 0x1Bでダッシュ (--) を表示。
16	非安全設定 チェックサム値 データ型: UINT 16ビット	0x70, 1, 12 (0x0C)	16ビット 非安全設定CRC値
17	安全設定 チェックサム値 データ型: UINT 16ビット	0x70, 1, 13 (0x0D)	16ビット 安全設定CRC値
18	有効警告エリア セット データ型: UINT 16ビット	0x70, 1, 14 (0x0E)	AZS_01 = 0 (0x00) ... AZS_70 = 69 (0x45) INVALID AZS = 32,768 (0x8000)
19	未使用		将来的に使用
20	未使用		将来的に使用
21	未使用		将来的に使用
22	未使用		将来的に使用
23	未使用		将来的に使用
24	検出距離レポート フォーマット データ型: UINT 16ビット	0x72, 1, 4	検出距離データの出力フォーマット
25	反射率レポート フォーマット データ型: UINT 16ビット	0x72, 1, 5	反射率データの出力フォーマット
26	検出距離レポート モード データ型: UINT 16ビット	0x72, 1, 6 (0x06)	REPORT_RANGE_ONLY = 0 (デフォルト) // アセンブリ102 検出距離レポートのみ REPORT_RANGE_AND_REFLECTIVITY = 1, // アセンブリ102 検出距離レポートと反射率レポート
27	ビーム数	0x72, 1, 17 (0x11)	レポート出力する計測ビーム数

表8で定義した共通の計測レポートヘッダフォーマットに加えて、入力アセンブリ102およびベンダー固有オブジェクト114には、選択した対象領域の検出距離計測データが含まれます。explicit TCP/IP要求・返信メッセージまたはimplicit UDP I/O接続を使用し、アプリケーションで指定した繰返し率によるOS32Cのモニタリングが可能です。

4.9 表9: 入力アセンブリ102およびベンダー固有オブジェクト114(最大1410バイト)

28 - 704	検出距離 (TOF) データ[] UINT 16ビット		検出距離計測データ、可変サイズ ビームレポートマスクを使ってサイズを選択(1) 配列サイズ = 計測レポートヘッダに含まれるビーム数、最大サイズ = 677
----------	--------------------------------	--	--

表8で定義した共通の計測レポートヘッダフォーマットに加えて、入力アセンブリ103には、選択した対象領域の反射率計測データが含まれます。explicit TCP/IP要求・返信メッセージまたはimplicit UDP I/O接続を使用し、アプリケーションで指定した繰返し率によるOS32Cのモニタリングが可能です。

4.10 表10 : 入力アセンブリ103およびベンダー固有オブジェクト116 (最大1410バイト)

28 - 704	反射率 (TOT) データ[] UINT 16ビット		反射率計測データ、可変サイズ。 ビームレポートマスクを使ってサイズを選択(1) 配列サイズ = 計測レポートヘッダに含まれるビーム数、最大サイズ = 677
----------	-------------------------------	--	--

表8で定義した共通の計測レポートヘッダフォーマットに加えて、ベンダー固有オブジェクト117には、選択した対象領域の検出距離計測データおよび反射率計測データが含まれます。UDP I/O 接続では1500バイト未満のメッセージに限定されるため、このアセンブリは、explicit TCP/IP要求・返信メッセージしか使用できません。OS32Cは、次のスキャン周期が終了した時点で、個々の要求に対する応答を行います。

4.11 表11 : ベンダー固有オブジェクト117 (最大2764バイト)

28 - 704	検出距離 (TOF) データ[] UINT 16ビット		検出距離計測データ、可変サイズ ビームレポートマスクを使ってサイズを選択 *1. *2. 配列サイズ = 計測レポートヘッダに含まれるビーム数、最大サイズ = 677
705-1381	反射率 (TOT) データ[] UINT 16ビット		反射率計測データ、可変サイズ ビームレポートマスクを使ってサイズを選択 *1. *2. 配列サイズ = 計測レポートヘッダに含まれるビーム数、最大サイズ = 677

*1. ビームレポート選択マスクの詳細は、4.4 表4 : 出力アセンブリ113およびベンダーオブジェクト115 (104バイト)、入力アセンブリ102、103用計測レポート設定を参照。

*2. ベンダー固有アセンブリレポートは、OS32Cのスキャン周期と同期しています。

表8で定義した共通の計測レポートヘッダフォーマットに加えて、入力アセンブリ104には、選択した対象領域の検出距離計測データが含まれます。explicit TCP/IP要求・返信メッセージまたはimplicit UDP I/O接続を使用し、アプリケーションで指定した繰返し率によるOS32Cのモニタリングが可能です。

4.12 表12: 入力アセンブリ104 (最大960バイト)

28 - 479	検出距離 (TOF) データ[] UINT 16ビット		検出距離計測データ、可変サイズ ビームレポートマスクを使ってサイズを選択 * 配列サイズ = 計測レポートヘッダに含まれるビーム数、最大サイズ = 452 (タグなし)
----------	-----------------------------------	--	--

* ビームレポート選択マスクの詳細は、表5: 出力アセンブリ114 (108バイト)、入力アセンブリ104、105用計測レポート設定を参照。

表8で定義した共通の計測レポートヘッダフォーマットに加えて、入力アセンブリ105には、選択した対象領域の反射率計測データが含まれます。explicit TCP/IP要求・返信メッセージまたはimplicit UDP I/O接続を使用し、アプリケーションで指定した繰返し率によるOS32Cのモニタリングが可能です。

4.13 表13: 入力アセンブリ105 (最大960バイト)

28 - 479	反射率 (TOT) データ[] UINT 16ビット		反射率計測データ、可変サイズ ビームレポートマスクを使ってサイズを選択 * 配列サイズ = 計測レポートヘッダに含まれるビーム数、最大サイズ = 452 (タグなし)
----------	----------------------------------	--	---

* ビームレポート選択マスクの詳細は、表5: 出力アセンブリ114 (108バイト)、入力アセンブリ104、105用計測レポート設定を参照。

入力アセンブリ106には、選択した対象領域の検出距離計測データが含まれます。explicit TCP/IP要求・返信メッセージまたはimplicit UDP I/O接続を使用し、アプリケーションで指定した繰返し率によるOS32Cのモニタリングが可能です。

4.14 表14: 入力アセンブリ106 (最大554バイト)

0 - 276	検出距離 (TOF) データ[] UINT 16ビット		検出距離計測データ、可変サイズ ビームレポートマスクを使ってサイズを選択 * 配列サイズ = 計測レポートヘッダに含まれるビーム数、最大サイズ = 277 (タグなし)
---------	-----------------------------------	--	--

* 表6: 出力アセンブリ115 (316バイト)、入力アセンブリ106~111用計測レポート設定を参照。

入力アセンブリ107には、選択した対象領域の反射率計測データが含まれます。explicit TCP/IP要求・返信メッセージまたはimplicit UDP I/O接続を使用し、アプリケーションで指定した繰返し率によるOS32Cのモニタリングが可能です。

4.15 表15: 入力アセンブリ107 (最大554バイト)

0 - 276	反射率 (TOT) データ[] UINT 16ビット		反射率計測データ、可変サイズ ビームレポートマスクを使ってサイズを選択 * 配列サイズ = 計測レポートヘッダに含まれるビーム数、最大サイズ = 277 (タグなし)
---------	----------------------------------	--	---

* 表6: 出力アセンブリ115 (316バイト)、入力アセンブリ106~111用計測レポート設定を参照。

入力アセンブリ108には、選択した対象領域の検出距離計測データが含まれます。explicit TCP/IP要求・返信メッセージまたはimplicit UDP I/O接続を使用し、アプリケーションで指定した繰返し率によるOS32Cのモニタリングが可能です。

4.16 表16: 入力アセンブリ108 (最大454バイト)

0 - 226	検出距離 (TOF) データ[] UINT 16ビット		検出距離計測データ、可変サイズ ビームレポートマスクを使ってサイズを選択 * 配列サイズ = 計測レポートヘッダに含まれるビーム数、最大サイズ = 227 (タグなし)
---------	-----------------------------------	--	--

* 表6: 出力アセンブリ115 (316バイト)、入力アセンブリ106~111用計測レポート設定 を参照。

入力アセンブリ109には、選択した対象領域の反射率計測データが含まれます。explicit TCP/IP要求・返信メッセージまたはimplicit UDP I/O接続を使用し、アプリケーションで指定した繰返し率によるOS32Cのモニタリングが可能です。

4.17 表17: 入力アセンブリ109 (最大454バイト)

0 - 226	反射率 (TOT) データ[] UINT 16ビット		反射率計測データ、可変サイズ ビームレポートマスクを使ってサイズを選択 * 配列サイズ = 計測レポートヘッダに含まれるビーム数、最大サイズ = 227 (タグなし)
---------	----------------------------------	--	---

* 表6: 出力アセンブリ115 (316バイト)、入力アセンブリ106~111用計測レポート設定 を参照。

入力アセンブリ110には、選択した対象領域の反射率計測データが含まれます。explicit TCP/IP要求・返信メッセージまたはimplicit UDP I/O接続を使用し、アプリケーションで指定した繰返し率によるOS32Cのモニタリングが可能です。

4.18 表18: 入力アセンブリ110 (最大358バイト)

0 - 178	検出距離 (TOF) データ[] UINT 16ビット		検出距離計測データ、可変サイズ ビームレポートマスクを使ってサイズを選択 * 配列サイズ = 計測レポートヘッダに含まれるビーム数、最大サイズ = 179 (タグなし)
---------	-----------------------------------	--	--

* 表6: 出力アセンブリ115 (316バイト)、入力アセンブリ106~111用計測レポート設定 を参照。

入力アセンブリ111には、選択した対象領域の反射率計測データが含まれます。explicit TCP/IP要求・返信メッセージまたはimplicit UDP I/O接続を使用し、アプリケーションで指定した繰返し率によるOS32Cのモニタリングが可能です。

4.19 表19: 入力アセンブリ111 (最大358バイト)

0 - 178	反射率 (TOT) データ[] UINT 16ビット		反射率計測データ、可変サイズ ビームレポートマスクを使ってサイズを選択 * 配列サイズ = 計測レポートヘッダに含まれるビーム数、最大サイズ = 179 (タグなし)
---------	----------------------------------	--	---

* 表6: 出力アセンブリ115 (316バイト)、入力アセンブリ106~111用計測レポート設定 を参照。

4.20 データリフレッシュ周期（予測パケット周期）

OS32Cのデータリフレッシュ周期はおよそ40（38.5～39.5）ミリ秒です。EtherNet/IP PLCネットワーク上でOS32Cを設定するときは、入力アセンブリデータの予測パケット周期を50ミリ秒以上の値に設定することをお勧めします。

4.21 有効警告エリアセットの選択

“モニタリングエリア”が複数の検知エリアセットに対して設定され、“EIPによる警告エリアの選択”が有効になっている場合、エリアセット選択入力の代わりにEtherNet/IPインターフェースを使って、有効警告エリアセットを変更することができます。この機能を有効にするときは、EtherNet/IPネットワークデバイスは、通常動作を開始する前に、あらかじめプログラムされた70個の警告エリアセットの候補の中から1つを選択しておく必要があります。エリアの設定と選択の詳細については、OS32Cユーザーマニュアルの「エリアセットの追加と削除」と「エリアセット選択の設定」の項を参照してください。EtherNet/IPを使用する有効警告エリアセットの切り替え時間は、1スキャン周期または約40ミリ秒です。出力アセンブリオブジェクト113、114、115、および表4のベンダー固有オブジェクト115はそれぞれ、有効警告エリアセットの定義に使用できます。

5. OS32C EDSファイルのインストール

OS32CのEDSファイルは、OS32C本体と同梱のOS32C設定ツールCDに格納されています。

以下の手順にしたがって、オムロンのNetwork ConfiguratorにEDSファイルをインストールします。

【EDSファイル】メニューで【インストール】を選択します。

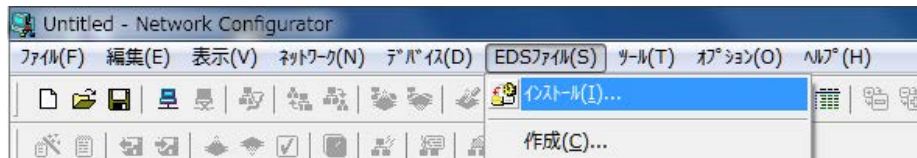


図5-1 EDSファイルのインストール

ダイアログ画面が立ち上がりますので、EDSファイルを選んで、[OK]をクリックします。Network Configuratorの左側にあるハードウェアリストにOS32Cが追加されます。

6. CJシリーズとの通信設定

ここでは、OS32Cとオムロン製PLC CJ2（あるいはCJ1）間のEtherNet/IP通信設定の方法について説明します。

6.1 EtherNet/IPネットワークをセットアップする

Network Configuratorを起動し、新規のEtherNet/IPネットワークを作成します。

ハードウェアリストから、EtherNet/IPネットワークに、OS32CとCJ2B-EIP21をドラッグ&ドロップして追加します。

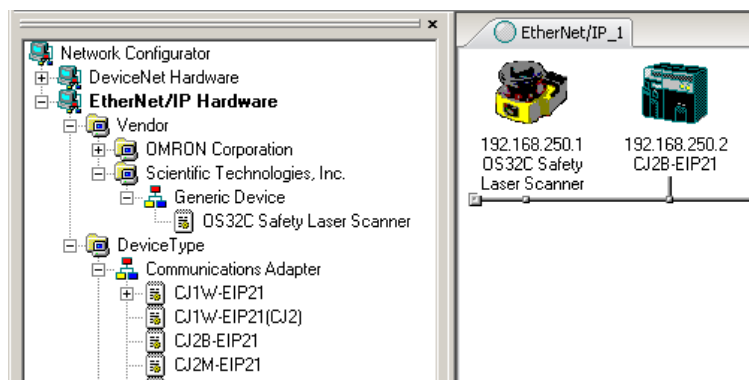


図6-1 ハードウェアリストからデバイスを選択

IPアドレスが、それぞれのデバイスと合致しているかどうか確認してください。IPアドレスを変更する場合は、デバイスのアイコンを右クリックし、[ノードアドレスの変更] を選びます。



図6-2 ノードアドレスの変更

ここでは、OS32CのIPアドレスを192.168.250.1、CJ2のIPアドレスを192.168.250.32とします。

【入力 - 受信】タブを選び、【新規】をクリックして新しいタグを作成します。

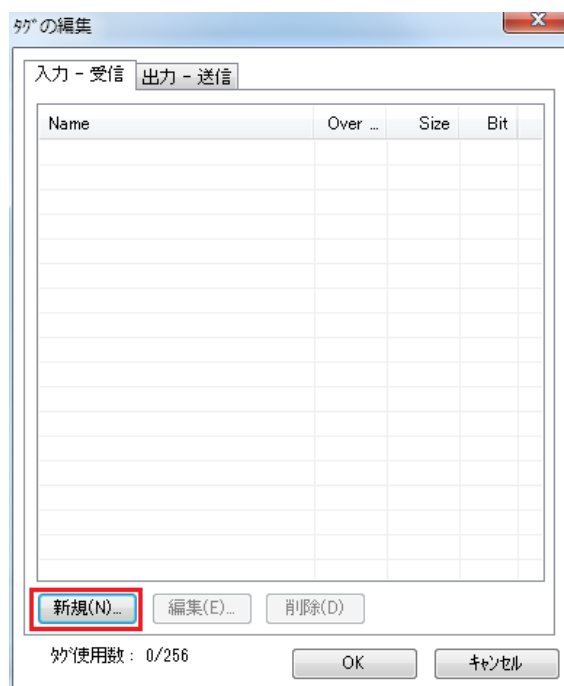


図6-4 タグの新規作成

【タグ名】にOS32Cデータを書き込むPLC内のメモリ場所を入力します。ここでは、【タグ名】にD0と入力し、CJ2のDMメモリ領域 0を指定します。

注意：このメモリ領域を指定するときは、必ず大文字を使用してください。ここでは図のように正確に”D0”と入力してください。小文字の”d”は無効になります。

【サイズ】に32バイトを入力し、【登録】ボタン、そして、【閉じる】ボタンをクリックします。



図6-5 タグの登録

[出力 - 送信] タブを選び、[新規] をクリックして新しいタグを作成します。

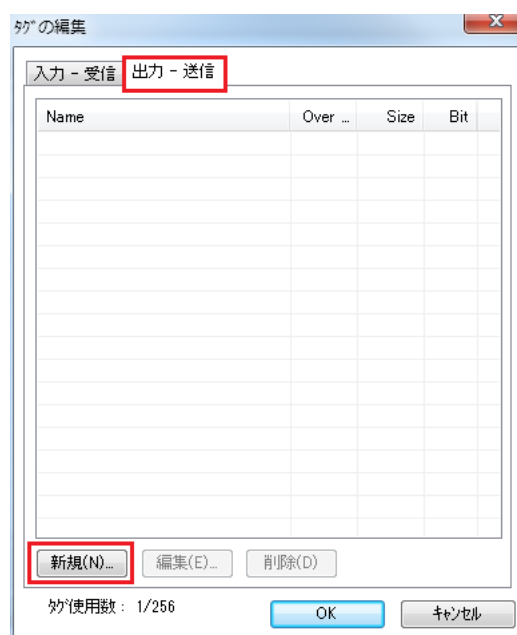


図6-6 タグの新規作成

[タグ名]にD100と入力し、CJ2のDMメモリ領域100を指定します。

メモ:D100のDは大文字でなければなりません。

[サイズ] に2バイトを入力し、[登録] ボタン、そして、[閉じる] ボタンをクリックします。



図6-7 タグの登録

[OK] をクリックすると、タグの作成が完成します。

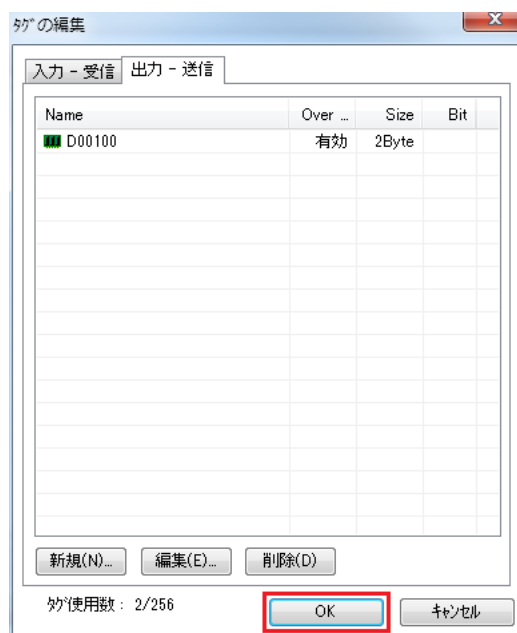


図6-8 [OK]をクリック

ダイアログ画面が表示されますので、[はい] をクリックして、新規作成したタグをタグセットとして登録します。

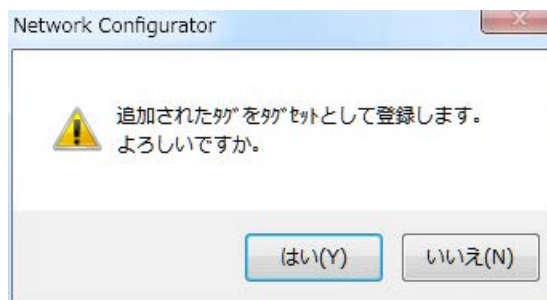


図6-9 タグの登録

[コネクション]タブを選択し、OS32Cを選んで、下向きの矢印ボタンをクリックしてください。これで、OS32CがCJ2のスレーブデバイスとして登録されます。

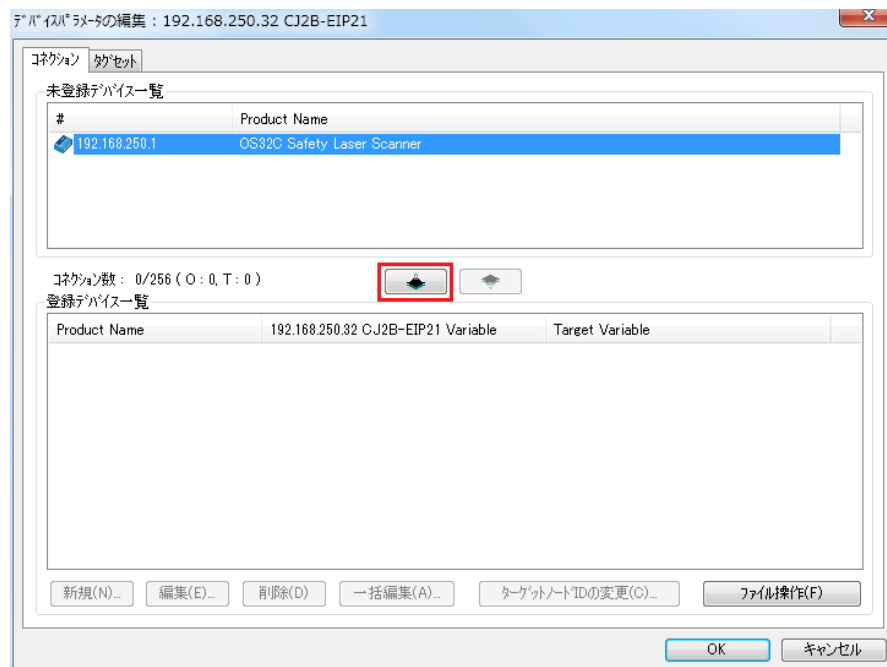


図6-10 CJ2へのOS32Cの登録

次に、登録デバイス一覧に登録されたOS32Cをダブルクリックして、コネクションの編集を行います。オリジネータデバイスの欄で、新規作成された入力および出力タグを選び、下の図のとおりを設定してください。



図6-11 コネクション割付ウィンドウ

画面左下の【詳細表示】ボタンをクリックして、[パケットインターバル(RPI)] を200msに設定します。
設定後、[登録] ボタン、[閉じる] ボタンをクリックします。

図6-12 パケット間隔の設定

「デバイスパラメータの編集」画面で、[OK]ボタンをクリックします。

Product Name	Variable	Target Variable
192.168.250.1 (#001) OS32C Safe...		
default_001 [Input]	D00000	Input_100
default_001 [Output]	D00100	Output_112

図6-13 登録されたOS32C

6.3 CJ2にEtherNet/IP設定をダウンロードする

以上の手順で、タグのセットアップが完成しました。次に、EtherNet/IP設定をCJ2にダウンロードします。USBケーブルを使って、パソコンをCJ2に接続してください。

Network Configuratorの[オプション]メニューで、[インターフェースの選択]を選び、[CJ2 USB/Serial Port]を選択します。

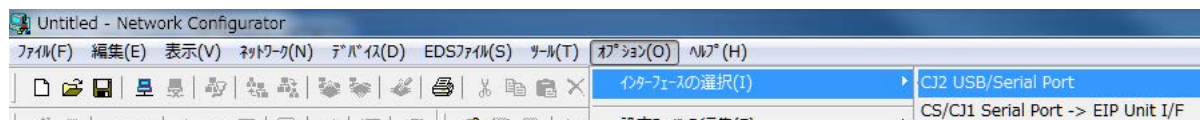


図6-14 インターフェースの選択

次に、[ネットワーク]メニューから[接続]を選択します。

[TCP:2]を選択し、[OK]をクリックすると、パソコンとCJ2が接続されます。

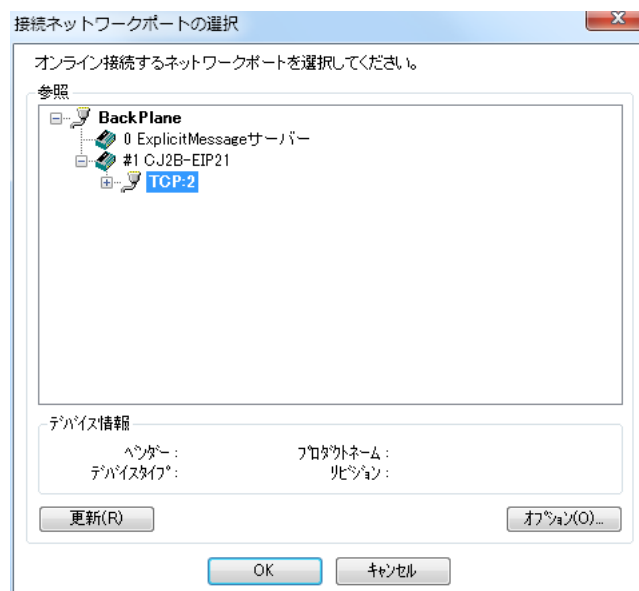


図6-15 ネットワークポートの選択

CJ2Bアイコンを右クリックし、[パラメータ]を選び、[ダウンロード]を選択します。



図6-16 デバイスへのダウンロード

表示された画面で、**[現在のモードでダウンロード]**をクリックすると、CJ2をプログラムモードに変更せずにモジュールをダウンロードすることができます。

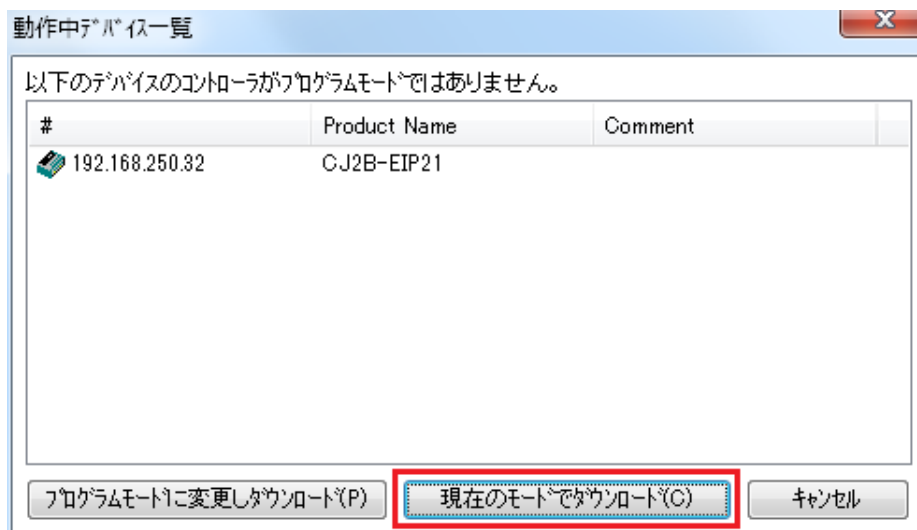


図6-17 カレントモードでのダウンロード

7. NJシリーズとの通信設定

ここでは、OS32Cとオムロン製PLC NJ5間のEtherNet/IP通信設定の方法について説明します。

7.1 NJ5 EthreNet/IPアダプタ設定

Sysmac Studioプログラミングツールを開き、該当する機械のプロジェクトを開きます。あるいは、新規のプロジェクトを作成します。下図のように、[構成・設定]メニューから[内蔵EtherNet/IPポート設定]をダブルクリックします。

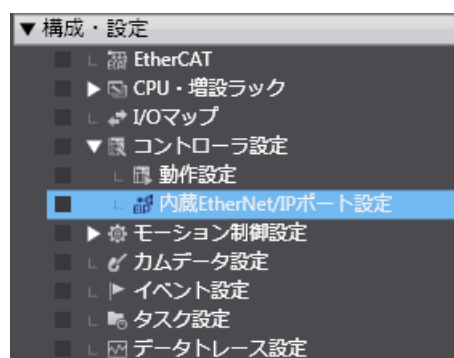


図7-1 内蔵EtherNet/IPポートの設定

表示されたウィンドウで、NJのIPアドレスを正しく設定してください。ここでは、NJを192.168.250.1に、OS32Cを192.168.250.7に設定します。

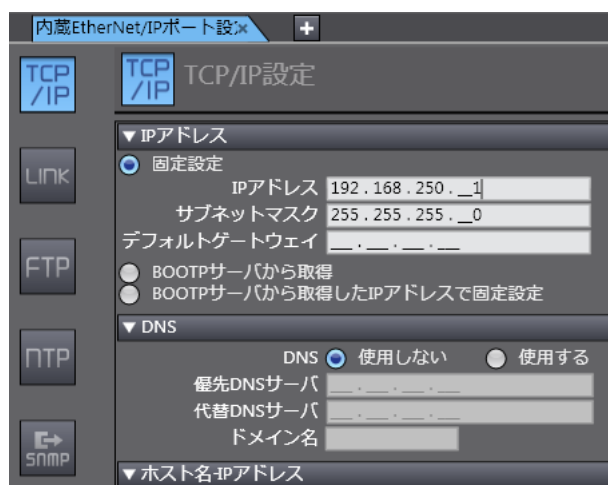


図7-2 IPアドレスの設定

7.2 NJにタグをセットアップする

NJシリーズは、タグベースのI/O構造をサポートしていますが、OS32CとNJ間のEtherNet/IP通信を正常に行うには、NJ内にタグを生成する必要があります。

メニュータブの[プログラミング]タブで、[データ]をクリックし、[データ型]を選びます。

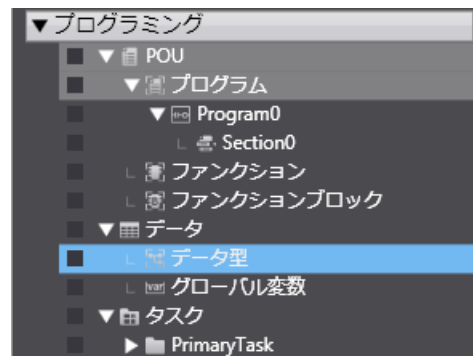


図7-3 データ型

表示されたウィンドウで、「名称」を右クリックし、[データ型新規作成]を選択します。

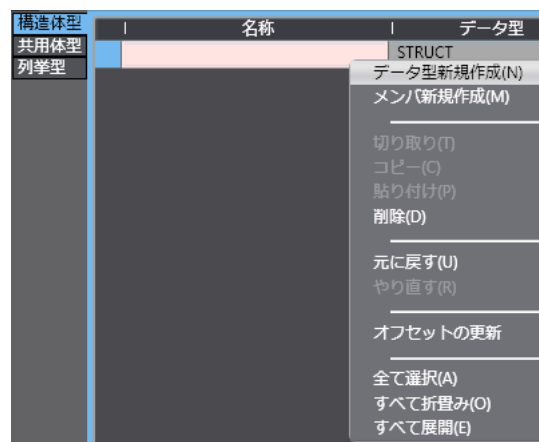


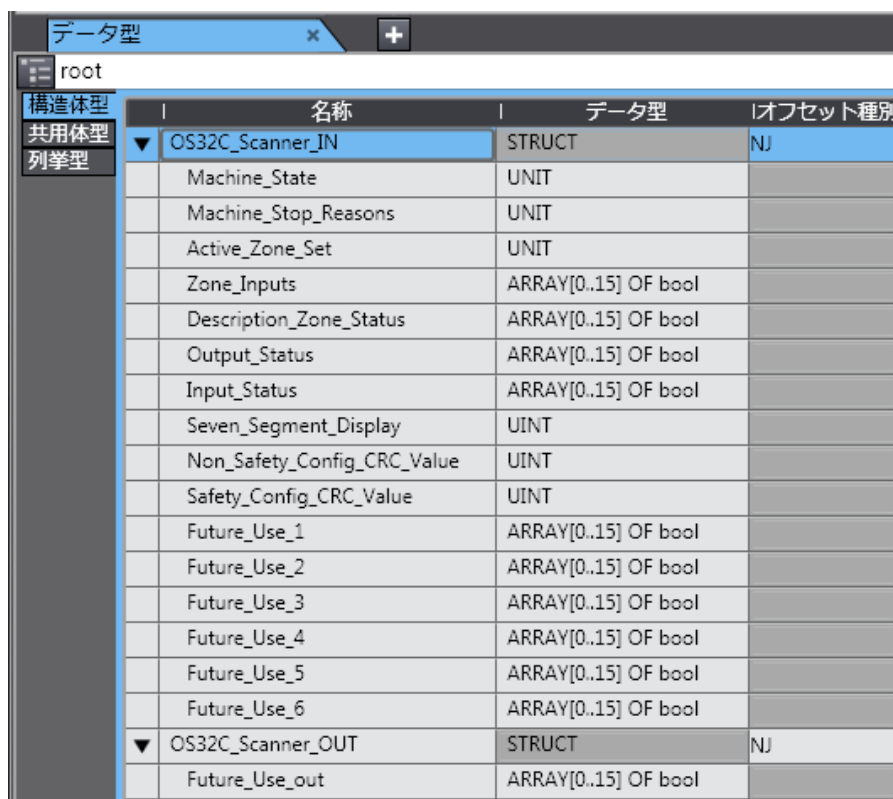
図7-4 データ型の新規作成

新規のデータ型として、「OS32C_Scanner_IN」と「OS32C_Scanner_OUT」の2つを作成し、いずれも[データ型]を「STRUCT」に指定します。生成されたタグは下図のように表示されます。

プログラミング						
データ型						
root						
構造体型	名称	データ型	オフセット種別	バイトオフセット	ビットオフセット	コメント
共用体型	OS32C_Scanner_IN	STRUCT	NJ			
列挙型	OS32C_Scanner_OUT	STRUCT	NJ			

図7-5 OS32CのIN構造体データ型とOUT構造体データ型

これで、ベースとなるタグが作成されました。次は、各タグにアセンブリI/Oデータを追加します。タグの名前を右クリックし、[メンバ新規作成]を選びます。下図のように、すべてI/Oデータが追加されるまで、この手順を繰り返してください。



名称	データ型	オフセット種別
OS32C_Scanner_IN	STRUCT	NJ
Machine_State	UNIT	
Machine_Stop_Reasons	UNIT	
Active_Zone_Set	UNIT	
Zone_Inputs	ARRAY[0..15] OF bool	
Description_Zone_Status	ARRAY[0..15] OF bool	
Output_Status	ARRAY[0..15] OF bool	
Input_Status	ARRAY[0..15] OF bool	
Seven_Segment_Display	UINT	
Non_Safety_Config_CRC_Value	UINT	
Safety_Config_CRC_Value	UINT	
Future_Use_1	ARRAY[0..15] OF bool	
Future_Use_2	ARRAY[0..15] OF bool	
Future_Use_3	ARRAY[0..15] OF bool	
Future_Use_4	ARRAY[0..15] OF bool	
Future_Use_5	ARRAY[0..15] OF bool	
Future_Use_6	ARRAY[0..15] OF bool	
OS32C_Scanner_OUT	STRUCT	NJ
Future_Use_out	ARRAY[0..15] OF bool	

図7-6 OS32C IN構造体メンバーとOUT構造体メンバ

7.3 タグをグローバル変数に設定する

以上の手順で、OS32CのEtherNet/IPタグとI/OデータがNJ内に生成されました。次に、NJがEtherNet/IP通信を行う個々のOS32Cのためのグローバル変数を生成する必要があります。

プロジェクトの作業スペースで、[グローバル変数]をクリックします。グローバル変数ウィンドウで右クリックし、[新規作成]を選ぶと、新しい変数が作成されます。

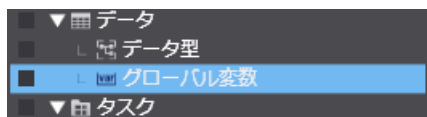


図7-7 グローバル変数



名称	データ型	初期値	割付先	保持	コンスタン	ネットワーク公開	コメント
	BOOL			☒	☒	非公開	

図7-8 グローバル変数の新規作成

作成した変数に名前をつけ、変数型から構造体名を選択します。また、タグに対して、ネットワークの入力か出力かを定義する必要があります。この設定を送信変数、受信変数についてそれぞれ1回ずつ行います。

OS32Cの送信・受信タグは下記の図のようになります。

グローバル変数							
名称	データ型	初期値	割付先	保持	コンスタン	ネットワーク公開	コメント
OS32C_Out_Node7	OS32C_Scanner_OUT			☐	☐	出力	
OS32C_In_Node7	OS32C_Scanner_IN			☐	☐	入力	

図7-9 OS32Cの送信・受信タグ

タグの作成では、OS32Cにより送信された情報は「入力」としての名前が付けられますので、注意が必要です。つまり、OS32Cから得られるデータは、NJ5によって受信されるため、NJ5への「入力」となります。逆に、OS32Cが受信する情報は、NJ5によって送信（出力）されたものであるため、ネットワーク公開のタイプは「出力」に設定します。

7.4 Network Configuratorにタグをエクスポートする

NJに作成されたタグをNetwork Configuratorにエクスポートし、使用することができます。これにより、タグのI/O構造が合致しているかを確認することができます。

メインウィンドウの[ツール]から[グローバル変数のエクスポート]を選び、[Network Configurator]を選択します。

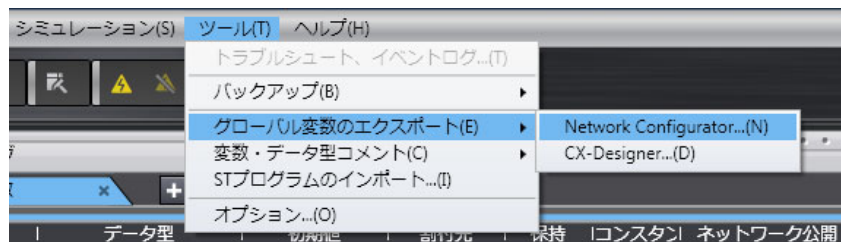


図7-10 グローバル変数のエクスポート

これにより、Excelワークシートのデータがcsvファイルで保存され、Network Configuratorからインポートできるようになります。

7.5 EtherNet/IPネットワークを設定する

NJへのEthernet /IP接続を設定するには、Network Configuratorを起動し、新規のEtherNet/IPプログラムを作成します。

ハードウェアリストから、EtherNet/IPネットワークに、NJ5とOS32Cをドラッグ＆ドロップします。IPアドレスがハードウェアのIP設定と同じになっているか確認してください。

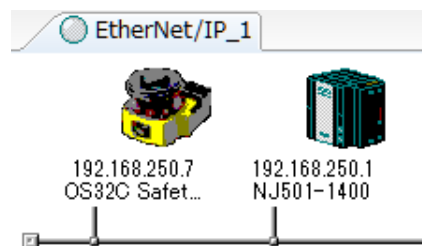


図7-11 NJ5とOS32CをEtherNet/IPネットワークへ追加

NJ5をダブルクリックし、[タグセット]タブを選びます。[ファイル操作]ボタンをクリックし、[ファイルからインポート]を選択してください。

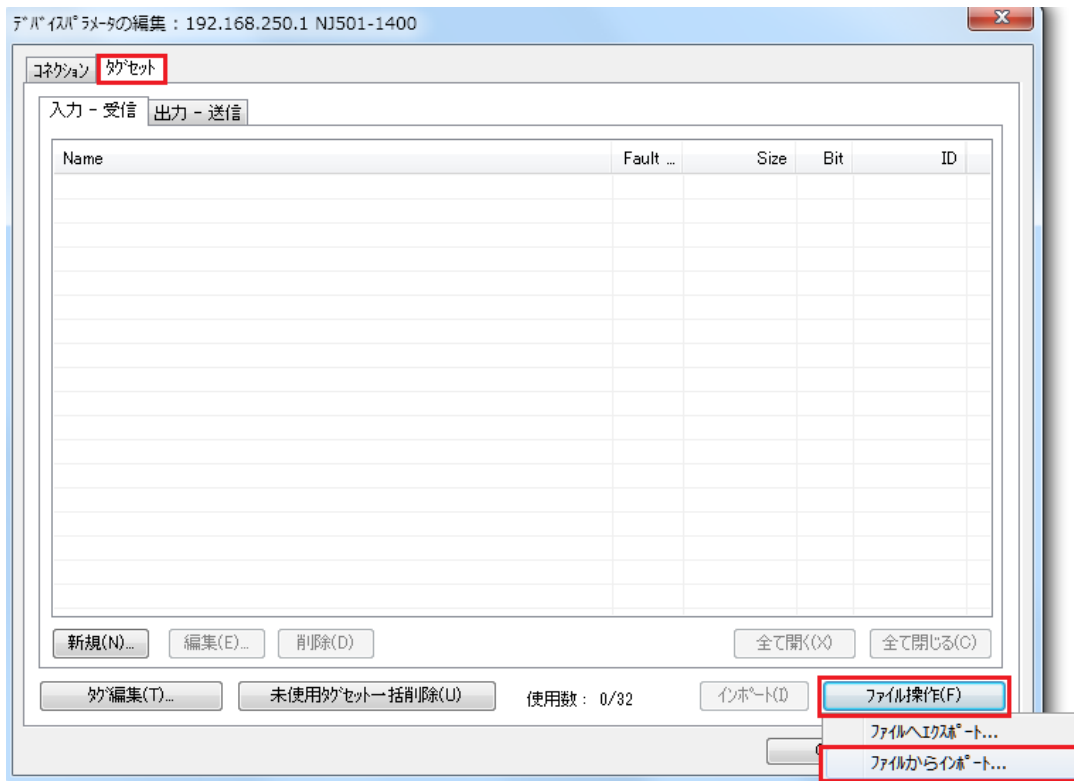


図7-12 タグセットをファイルからインポート

Sysmac Studioからエクスポートされた、ネットワークタグが含まれているcsvファイルを探し、[OK]ボタンを押します。ダイアログが表示されたら、[はい]をクリックし、シンボルをインポートします。これで、タグが[入力 - 受信]タブおよび[出力 - 送信]タブにそれぞれ追加されます。追加されると、[入力 - 受信]タブと[出力 - 送信]タブが下図のように表示されます。

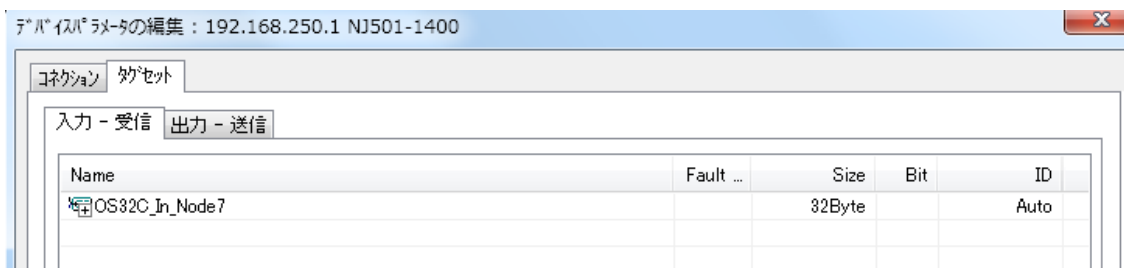


図7-13 入力 - 受信タブ

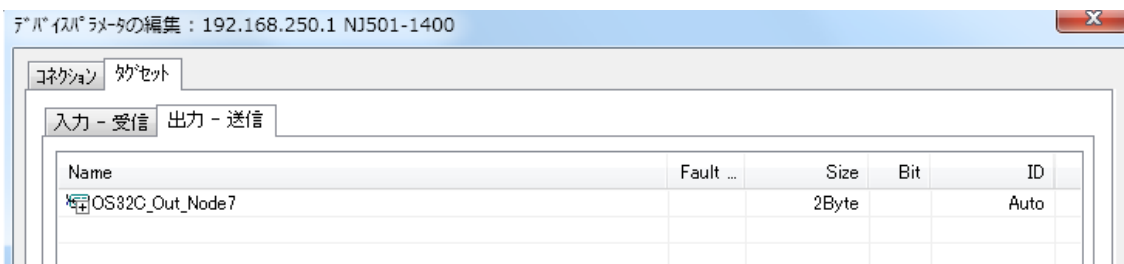


図7-14 出力 - 送信タブ

[コネクション]タブを選びます。OS32Cモジュールが未登録デバイス一覧の下に表示されます。OS32Cモジュールを選択し、**下方向の矢印**をクリックすると、OS32Cが未登録デバイス一覧から登録デバイス一覧へと移動します。

OS32Cが登録デバイス一覧に表示されたら、OS32Cをダブルクリックしてください。コネクション割付ウィンドウが表示されます。左側のタグ名に合わせて、右側の入力および出力アセンブリを設定します。ここでは、個々の接続において、タグは1つのみとしています。RPI値には**200**を指定してください。設定が完了すると、下図のようになります。

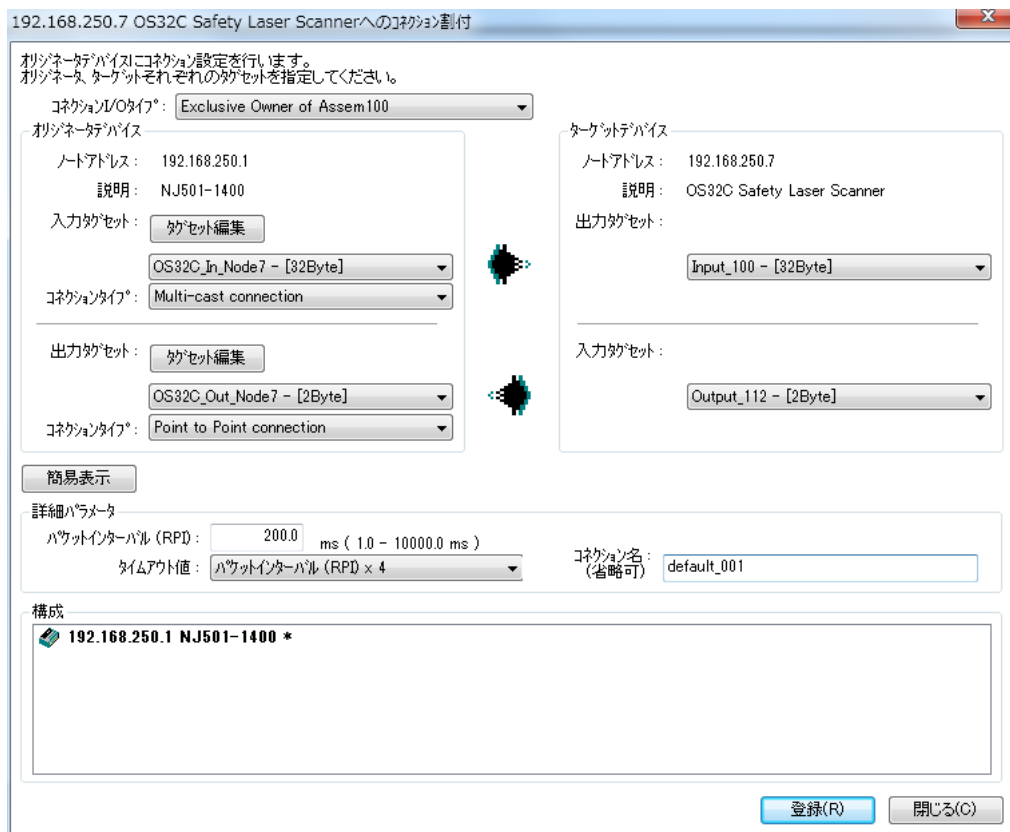


図7-15 コネクション割付ウィンドウ

ウィンドウ下の[登録]ボタンを押した後、[閉じる]ボタンを押します。最後に、「デバイスパラメータの編集」ページで[OK]をクリックしてください。

ネットワークが下図のように表示されます。赤線で囲んだ部分の矢印は、ノードがNJ5に登録されていることを示しています。

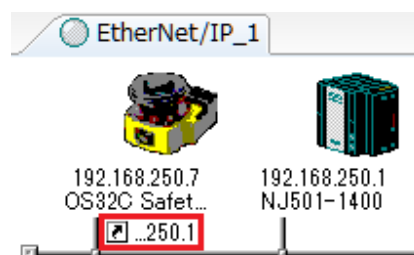


図7-16 NJ5に登録されたOS32C

7.6 EtherNet/IP設定をNJにダウンロードする

NJ5をUSBケーブルでパソコンに接続します。

Network Configuratorで、[オプション]のプルダウンメニューをクリックし、[インタフェースの選択]を選び、[NJ Series USB Port]を選択してください。

次に、[ネットワーク]メニューから[接続]を選択します。

[TCP:2]をクリックし、[OK]を押します。

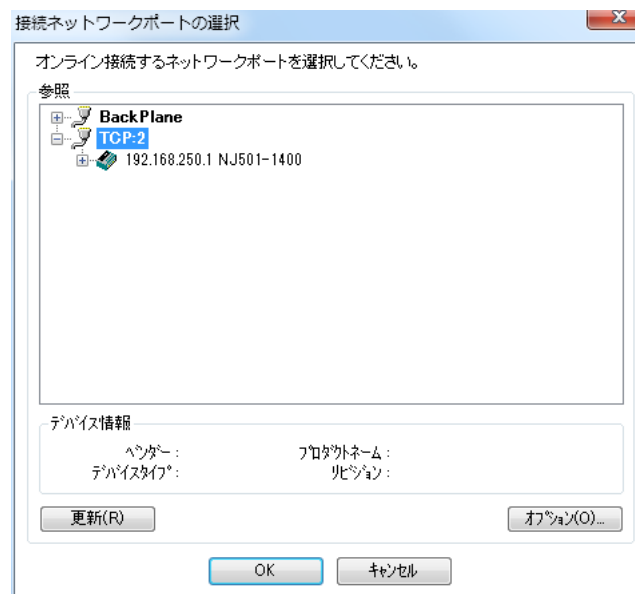


図7-17 ネットワークポートの選択

[既存のネットワーク]を選び、[OK]をクリックしてください。

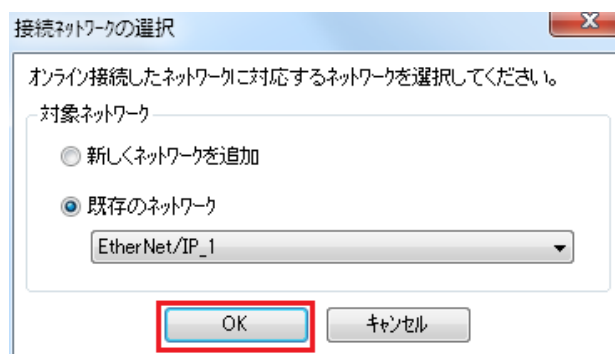


図7-18 既存のネットワーク

NJ5アイコンを右クリックし、[パラメータ]を選び、[ダウンロード]を選択します。



図7-19 NJ5へのダウンロード

[現在のモードでダウンロード]をクリックすると、NJ5をプログラムモードに変更せずにダウンロードすることができます。

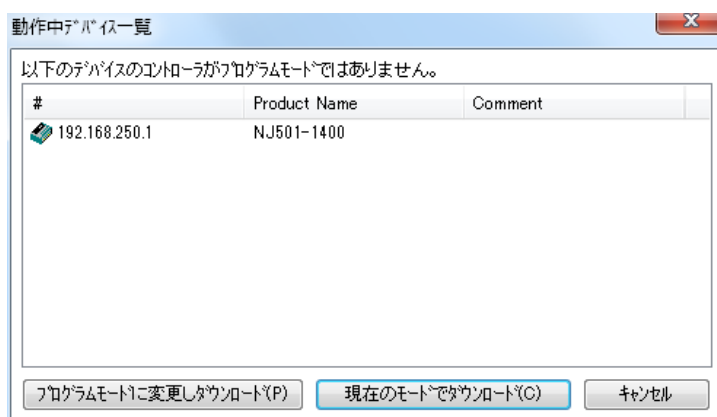


図7-20 カレントモードでのダウンロード

ダウンロードが完了したら、[OK]をクリックします。これでセットアップは完了しました。

8. 複数台のOS32C、PLCのセットアップ

EtherNet/IP対応OS32Cは、複数ユニットを使ったポーリングが可能です。複数のEtherNet/IPマスターデバイスから1台のOS32Cをポーリングしたり、1台のEtherNet/IPマスターデバイスから複数のOS32Cをポーリングすることも可能です。

8.1 1台のPLCから複数のOS32Cをポーリングする

1台のマスターデバイスから複数のOS32Cをポーリングします。

Network Configuratorを開き、EtherNet/IPネットワークに、必要台数分のOS32Cを追加します。ここでは、1台のCJ2コントローラと3台のOS32Cを想定します。



図8-1 1台のPLCと複数のOS32C

CJ1W-EIP21アイコンをダブルクリックし、[タグセット] タブを選びます。さらに、[入力 - 受信] タブを選び、[タグ編集] ボタンをクリックしてください。

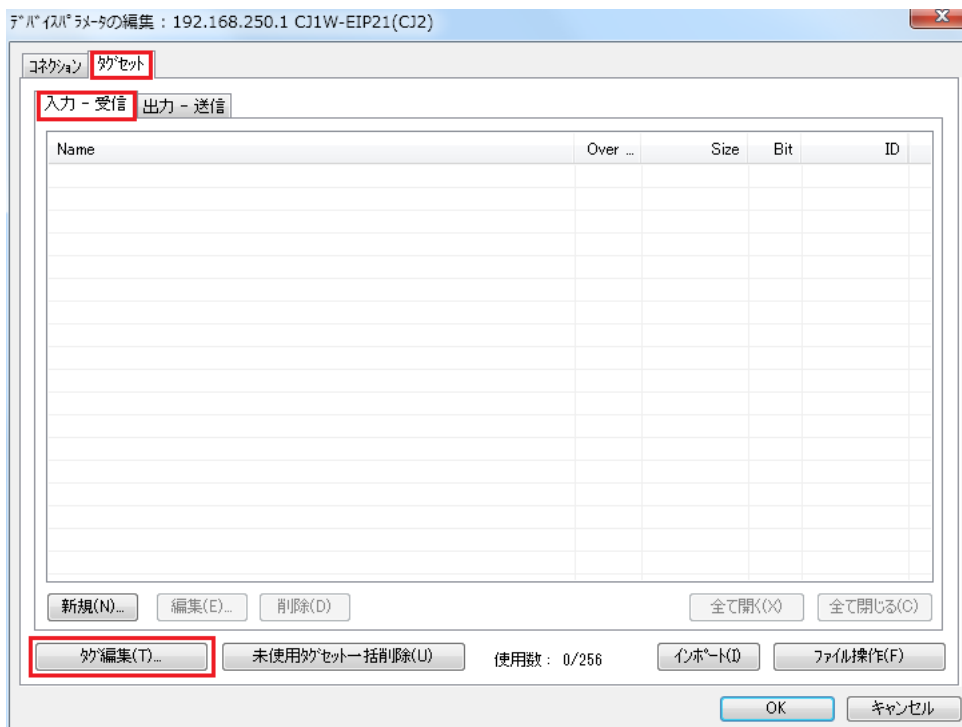


図8-2 タグの編集

[タグ名] にD0を入力してください。ここで指定されたメモリ場所にしがって、PLCに書き込みが行われます。[サイズ] には32バイトを指定します。この値は、OS32Cの[ターゲット入力]バイト数と合わせる必要があります。この手順は、このマスターに登録するOS32Cの台数分だけ繰り返します。メモリ場所が重複しないよう注意してください。

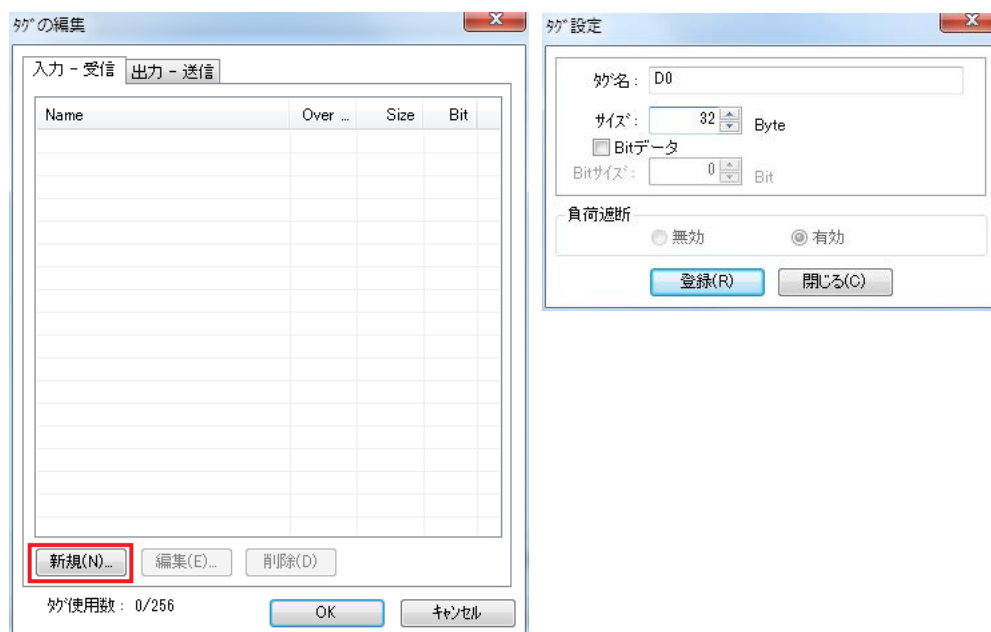


図8-3 [入力]タグの作成

[登録]をクリックするとタグが作成されます。以上の手順をすべてのOS32Cに対して行ってください。

[出力 - 送信] タブを選び、[新規] をクリックして新しいタグを作成します。

[タグ名] に **D100**を入力してください。ここで指定されたメモリ場所にしがって、PLCに書き込みが行われます。PLCでは、D100はDM100となります。[サイズ] には**2バイト**を指定します。この値は、OS32Cの[ターゲット出力]バイト数と合わせる必要があります。

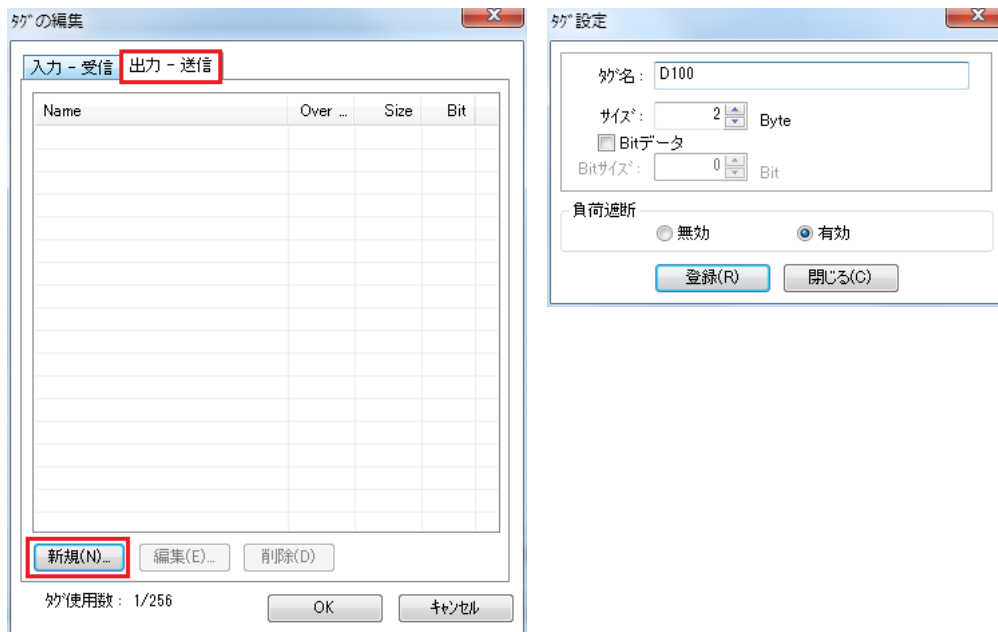


図8-4 [出力]タグの作成

[登録]をクリックするとタグが作成されます。

すべてのタグ作成が完了すると、タグウィンドウは、下図のようになります。

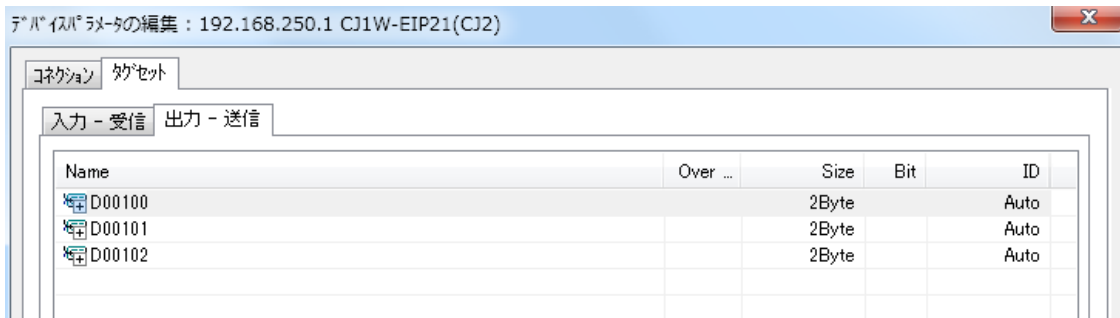


図8-5 作成が完了した[出力]タグ

[コネクション]タブを選択し、OS32Cを選んで、**下向きの矢印** ボタンをクリックしてください。下図のように、未登録デバイスリストから登録デバイスリストへと移動します。このマスタに登録するすべてのOS32Cに対して、この手順を繰り返します。

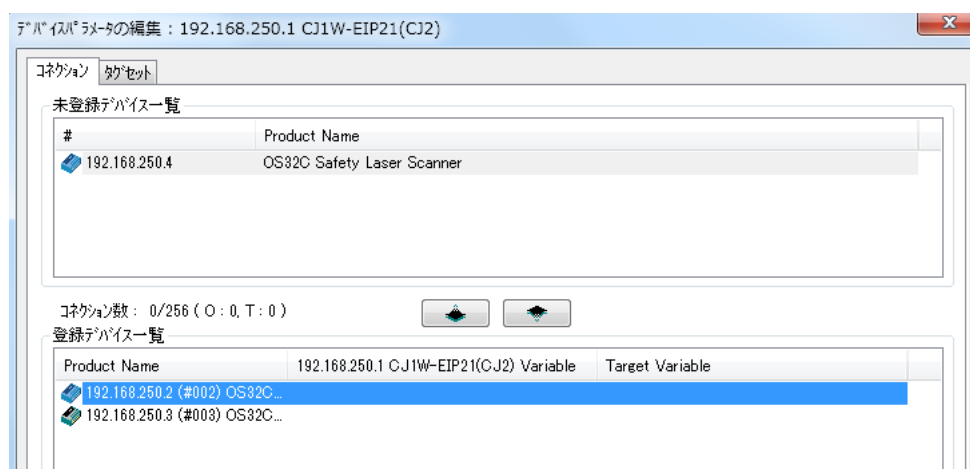


図8-6 PLCへ各OS32Cを登録

登録デバイスリストでそれぞれのOS32Cをダブルクリックし、OS32CのI/O場所を設定します。3台のOS32Cに対して、それぞれ入力タグと出力タグを選んで設定してください。OS32Cからのデータが書き込まれるPLCのDMの場所を選択します。この手順は、すべてのOS32Cに対して行います。

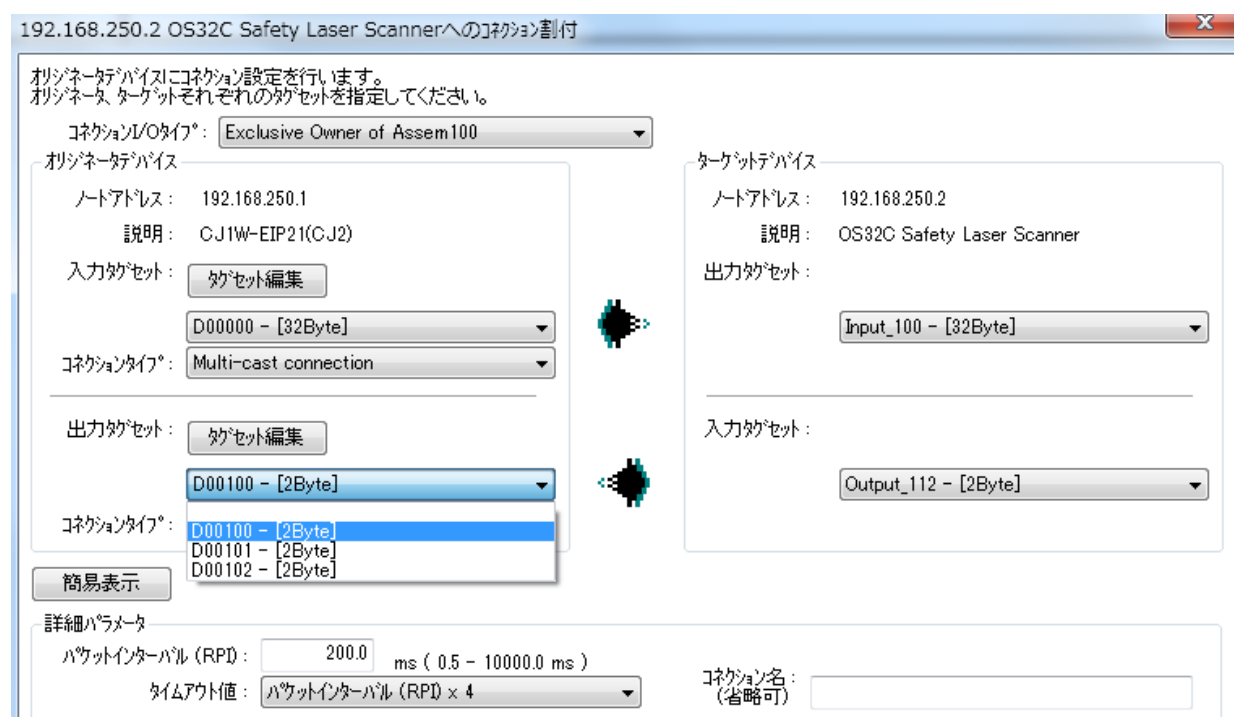


図8-7 コネクション割付ウィンドウ

すべてのOS32Cの設定が完了すると、画面は下図のようになります。



図8-8 OS32Cの設定完了

これで、3台のOS32Cが、CJ2に登録されました。

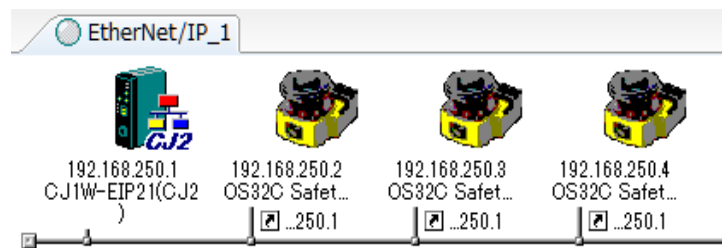


図8-9 登録完了

8.2 複数のPLCから1台のOS32Cをポーリングする

複数のデバイスから1台のOS32Cをポーリングするための設定は、1台のデバイスによる複数のOS32Cのポーリング設定とほとんど同じです。この設定のポイントは、複数デバイスのうち1台をOS32Cの[Exclusive Owner]とし、他のデバイスは[Listen only]あるいは[Input Only]として設定をします。

「Input Only」設定例

192.168.250.2 OS32C Safety Laser Scannerへの接続割付

オリジネータデバイスに接続設定を行います。
オリジネータ、ターゲットそれぞれのデータセットを指定してください。

接続元/タイプ: Input Only of Assem100

オリジネータデバイス

ノードアドレス: 192.168.250.3
説明: C-J1W-EIP21(CJ2)

入力データセット: データセット編集
D00200 - [32Byte]

接続タイプ: Multi-cast connection

出力データセット: データセット編集

接続タイプ: Point to Point connection

ターゲットデバイス

ノードアドレス: 192.168.250.2
説明: OS32C Safety Laser Scanner

出力データセット: Input_100 - [32Byte]

入力データセット:

簡易表示

詳細パラメータ

パケットインターバル (RPD): 200.0 ms (0.5 - 10000.0 ms)
タイムアウト値: パケットインターバル (RPD) × 4

接続名: (省略可) default_001

図8-10 「Input Only」設定例

「Listen Only」設定例

192.168.250.2 OS32C Safety Laser Scannerへの接続割付

オリジネータデバイスに接続設定を行います。
オリジネータ、ターゲットそれぞれのデータセットを指定してください。

接続元/タイプ: Listen Only of Assem100

オリジネータデバイス

ノードアドレス: 192.168.250.3
説明: C-J1W-EIP21(CJ2)

入力データセット: データセット編集
D00200 - [32Byte]

接続タイプ: Multi-cast connection

出力データセット: データセット編集

接続タイプ: Point to Point connection

ターゲットデバイス

ノードアドレス: 192.168.250.2
説明: OS32C Safety Laser Scanner

出力データセット: Input_100 - [32Byte]

入力データセット:

簡易表示

詳細パラメータ

パケットインターバル (RPD): 200.0 ms (0.5 - 10000.0 ms)
タイムアウト値: パケットインターバル (RPD) × 4

接続名: (省略可)

図8-11 「Listen Only」設定例

「Input Only」、「Listen Only」のうち、どちらの設定も可能ですが、両者には以下のような違いがあるため、注意が必要です。

1. 「Input Only」設定例:この設定では、デバイスは、RPI設定レートに基づいて、OS32Cへのポーリングメッセージを生成するため、多くのデバイスがOS32Cのデータを要求している場合、問題となる可能性があります。
2. 「Listen Only」:この設定では、デバイスは、ネットワーク上のマスタによって生成されたメッセージをリスンするのみで、自身がメッセージを生成することはありません。ネットワーク上のメッセージ生成を制限することはできますが、マスタデバイスがポーリングを停止した場合、他のデバイスのポーリングも停止となります。

9. パソコンベースのデバイスとの通信を設定する

ここでは、OS32C とパソコンベースのデバイス間との通信設定について説明しています。また、単一の EtherNet/IP サブネットやネットワークにおける OS32C との通信に必要な EtherNet/IP の仕様についても述べています。標準 Ethernet 型のネットワークを使用する場合も、このマニュアルで提供している通信フォーマットを使ってサブネット・ネットワーク通信が可能です。

9.1 EtherNet/IP コマンドプロトコル

電子データグラムコマンドのペイロードはすべて、24 バイトの固定長を持つヘッダとそれに続く任意のデータ部分からなります。データグラムの全ペイロード長は、TCP/IP メッセージでは 65535 バイト、UDP/IP メッセージでは 1500 バイトに制限されています。下の表 10 は、EtherNet/IP コマンドの基本構造の概要です。

9.1.1 表 10 : EtherNet/IP データグラムヘッダのコマンドフォーマット

構造	フィールド名	データ型	データサイズ	フィールド値
カプセル化ヘッダ	Command	UINT	2 バイト	カプセル化コマンド番号
	Length	UINT	2 バイト	カプセル化ヘッダに続く、メッセージのコマンド固有データ部分のバイト長
	Session Handle ID	UDINT	4 バイト	設定やランタイムモニタリングに使用するセッション ID
	Status	UDINT	4 バイト	返信メッセージに使用するステータスコード
	Sender Context Data	8 オクテット配列	8 バイト	カプセル化コマンドの送信者固有の情報
	Command options	UDINT	4 バイト	任意のコマンドフラグ
	Command data	0～65511 オクテット配列	0～65511 バイト	特定のコマンドにのみ要求される、メッセージのカプセル化されたデータ部分

9.2 EtherNet/IP コマンドリスト

EtherNet/IP では、下の表 11 のような基本コマンドを使って、全スキャナデータにアクセスすることができます。

9.2.1 表 11 : EtherNet/IP コマンドリスト

コマンド名	コマンドコード	説明
NOP	0x0000	TCP 通信中に TCP 接続検証のために使用する無操作コマンド（TCP のみを使用して送信可能）
List Services	0x0004	EtherNet/IP サービスが使用できる OS32C のリストを提供（UDP と TCP のいずれかを使用して送信可能）
List Identity	0x0063	OS32C の EtherNet/IP 識別、ベンダー ID、デバイス ID、シリアルナンバーなどの情報リストを提供（UDP と TCP のいずれかを使用して送信可能）
List Interfaces	0x0064	OS32C の EtherNet/IP アセンブリ、入力・出力オブジェクトインタフェースのリストを提供（UDP と TCP のいずれかを使用して送信可能）
Register Session	0x0065	OS32C との通信セッションを開始、確立（TCP のみを使用して送信可能）
Un-Register Session	0x0066	確立された OS32C との通信セッションを終了（TCP のみを使用して送信可能）
SendRRData	0x006F	サブコマンドおよび任意データとともに要求・応答コマンドを OS32C に送信（TCP のみを使用して送信可能）

OS32Cとのコマンド要求が正常に確立された場合、Statusフィールドは「0」になり、正常に確立されなかった場合、Statusフィールドには、下記のいずれかのエラーコードが含まれます。

9.2.2 表12 : EtherNet/IP状態エラーコードリスト

エラーコード	説明
0x0000	コマンド要求エラーなし
0x0001	無効なコマンドを使った要求
0x0002	対象デバイスのメモリ不足
0x0003	不正なデータを使用した要求
0x0064	無効なセッションハンドルを使用した要求
0x0065	無効なコマンドの長さの要求
0x0069	サポートされていないプロトコルのバージョンを使った要求

9.3 EtherNet/IPコマンド固有データ

下の表13のように、OS32Cが提供するコマンド固有データは、CIP共通パケットフォーマットに準拠する必要があります。

9.3.1 表13 : EtherNet/IP共通パケットフォーマット (CPF)

フィールド名	データ型				説明
Item count	UINT				準拠項目数
Item #1	項目の構造	フィールド名	データ型	説明	1番目の共通パケットフォーマット (CPF) 項目
		Type ID	UINT	カプセル化項目の種類	
		Length	UINT	Dataフィールドのバイト長	
		Data	Variable	データ (長さ>0)	
Item #2	項目の構造 (上記参照)				2番目のCPF項目
-----	-----				-----
Item #n	項目の構造 (上記参照)				n番目のCPF項目

9.4 EtherNet/IPコマンド

9.4.1 List Identityコマンド

List Identityコマンドは、発信元が使用することで、OS32Cの特定および識別ができます。コマンドは、TCPあるいはUDPを使ったユニキャストメッセージや、あるいは、UDPによるブロードキャストメッセージとして送信しますが、後者の場合、セッションを確立する必要がありません。返信は常にユニキャストメッセージとして返されます。

ブロードキャストメッセージとして受信した場合、受信デバイス側では、擬似ランダム遅延時間の後、EtherNet/IP標準の2-4.2.3セクションの指定に従って応答を返します。この応答の遅延によって、送信されたARP要求とネットワーク上の対象デバイスからのList Identity応答が確実にやりとりされます。

9.4.2 表14 : List Identityコマンド要求

データグラム	フィールド名	フィールド値	データ型	データサイズ	備考
EtherNet/IPヘッダ	Command	0x0063	UINT	2バイト	List Identityコマンド
	Length	0	UINT	2バイト	コマンド固有データの長さ
	Session Handle ID	任意の値	UDINT	4バイト	任意（対象デバイス側で無視）
	Status	0	UDINT	4バイト	0
	Sender Context Data		UINT	2バイト	最大応答遅延時間（ミリ秒）
		0	ARRAY[6]	6バイト	受信側はReserved(保留)フィールドを無視、値は「0」
	Command options	0	UDINT	4バイト	0

9.4.3 表15 : List Identityコマンド応答

データグラム	フィールド名	フィールド値	データ型	データサイズ	備考
EtherNet/IPヘッダ	Command	0x0063	UINT	2バイト	List Identityコマンド
	Length		UINT	2バイト	コマンド固有データの長さ
	Session Handle ID		UDINT	4バイト	任意（受信側で無視）
	Status	0	UDINT	4バイト	正常に行われた場合のエラー状態は0x0000
	Sender Context Data	Array[8]	オクテット配列	8バイト	オリジナル要求の値、長さ8 最大応答遅延時間（ミリ秒）
	Command options	0	UDINT	4バイト	
コマンド固有データ	Item Count	1	UINT	2バイト	
	Item ID	0x0C	UINT	2バイト	CIP識別オブジェクト
	Item Length	2	UINT	2バイト	項目のバイト数
	オクテット配列				表16 : EtherNet/IP識別オブジェクトパラメータを参照.

9.4.4 表16 : EtherNet/IP識別オブジェクトパラメータ

パラメータ名	データ型	説明
Encapsulation Protocol Version	UINT	サポートされているカプセル化プロトコルのバージョン（Register Session応答によっても返される）
Socket Address	下記の構造体	
	INT	sin_family（ビッグエンディアン）
	UINT	sin_port（ビッグエンディアン）
	UDINT	sin_addr（ビッグエンディアン）
	ARRAY[8]	UINT8 [8] sin_zero（長さ8）（ビッグエンディアン）
Vendor ID	UINT	デバイス製造元のベンダー ID
Device Type	UINT	製品のデバイス種類
Product Code	UINT	デバイス種類ごとの割り当て製品コード
Revision	ARRAY[2]	デバイス改定
Status	WORD	デバイスの現在の状態
Serial Number	UDINT	デバイスのシリアルナンバー
Product Name	ARRAY[32]	デバイスの通称
State	UINT8	デバイスの現在の状況

9.4.5 Register Sessionコマンド

Register sessionコマンドは、TCPポートを使用しているOS32CからTCPセッションのハンドルを取得するだけの単一ステップのコマンドです。取得したセッションハンドルは、それ以後のすべてのTCPおよびUDP I/O通信に使用することができます。List IdentityまたはList Interfaceコマンドを使った標準UDP通信では、セッションハンドルは必要ありません。

9.4.5.1 必要順序：

- ・ Register Sessionコマンドを使ってOS32Cからのセッションハンドルを要求します。

9.4.6 表17 : Register Sessionコマンド要求

データグラム部分	フィールド名	フィールド値	データ型	データサイズ	備考
EtherNet/IPヘッダ	Command	0x0065	UINT	2バイト	セッションハンドルの要求
	Length	4	UINT	2バイト	コマンド固有データの長さ
	Session Handle ID	0	UDINT	4バイト	任意（OS32C側で無視）
	Status	0	UDINT	4バイト	
	Sender Context Data	Array[8]	オクテット配列	8バイト	任意、長さ8
	Command options	0	UDINT	4バイト	0
コマンド固有データ	Protocol Version	1	UINT	2バイト	
	Option flags	0	UINT	2バイト	

9.4.7 表18 : Register Sessionコマンド応答

テレグラム・データグラム部分	フィールド名	フィールド値	データ型	データサイズ	備考
EtherNet/IPヘッダ	Command	0x0065	UINT	2バイト	セッションハンドルの要求
	Length	4	UINT	2バイト	コマンド固有データの長さ
	Session Handle ID		UDINT	4バイト	OS32Cから返されるセッションハンドルのID
	Status	0	UDINT	4バイト	正常に行われた場合のエラー状態は0x0000
	Sender Context Data	Array[8]	オクテット配列	8バイト	要求の値、長さ8
	Command options	0	UDINT	4バイト	0
	Protocol Version		UINT	2バイト	サポートされている場合はRegister Session要求のバージョン。要求バージョンが未サポートの場合、サポートしている最大バージョンを格納。
コマンド固有データ	Option flags		UINT	2バイト	サポートされている場合はRegister Session要求の任意フラグ。要求された任意フラグが未サポートの場合、サポートされている任意フラグを格納。

ヘッダのSession Handleフィールドには、OS32Cが生成する識別子が格納されます。クライアントのアプリケーションがこれを保存し、それ以後のすべてのコマンド要求について、ヘッダのSession Handleフィールドに挿入されます。このフィールドは、Statusフィールドが「0」の場合のみ、有効になります。クライアントのアプリケーションが正常にOS32Cに登録されると、Statusフィールドは「0」となり、正常に登録されなかった場合は、次のようなエラーコードがStatusフィールドに返されます。

- ・エラーコード 0x0001 : クライアントのアプリケーションが同一のTCP接続において1回以上アクティブなセッションを登録しようと試みた場合
- ・エラーコード 0x0002 : OS32C側にクライアントのアプリケーションを登録するための十分なリソースがない場合
- ・エラーコード 0x0069 : プロトコルバージョンやオプションが、以下のように不一致である場合

クライアントのアプリケーションが正常に登録された場合、Protocol Versionフィールドには要求されたバージョンが格納されますが、OS32C側で要求バージョンをサポートしていない場合、

- ・セッションは確立しない
- ・Statusフィールドは「サポートされていないカプセル化プロトコル」(0x0069) となる
- ・OS32Cは、サポートされている最大のバージョンをProtocol Versionフィールドに返す

9.4.8 Un-Register Sessionコマンド

クライアントのアプリケーションあるいはOS32Cが、TCPポートを使ってこのコマンドを送信し、セッションを終了させます。このコマンドを受け取った場合、受信側は基礎になるTCP/IP接続の終了を開始します。また、クライアントのアプリケーションとOS32C間の接続が終了した場合もセッションは終了します。受信側は、セッションの終了に必要なあらゆる処理を実行します。このコマンドに対する応答はありません。

9.4.8.1 必要順序 :

- ・ Register Sessionコマンド (セクション9.4.5参照) を使ってOS32Cからのセッションハンドルを要求後
- ・ 現状のセッションハンドルを用いてOS32Cからログオフ

9.4.9 表19 : Un-Register Sessionコマンド要求

テレグラム・データグラム部分	フィールド名	フィールド値	データ型	データサイズ	備考
EtherNet/IPヘッダ	Command	0x0066	UINT	2バイト	セッションの終了
	Length	0	UINT	2バイト	コマンド固有データの長さ
	Session Handle ID		UDINT	4バイト	現状のセッションハンドルID値
	Status	0	UDINT	4バイト	
	Sender Context Data	Array[8]	オクテット配列	8バイト	任意、長さ8（OS32C側で無視）
	Command options	0	UDINT	4バイト	0

受信側は、カプセル化ヘッダの予期しない値（無効なセッションハンドル、0以外の状態、0以外のオプション、あるいは追加コマンドデータなど）が原因で、Un-Register Sessionコマンドを拒否することはありません。いずれの場合も、TCP接続は終了します。

9.4.10 SendRRDataコマンド

SendRRDataコマンドは、TCPポートを使って発信元と対象となるOS32Cとの間でカプセル化した要求・応答パケットを交換するためのコマンドで、発信元が送信します。実際の要求・応答パケットは、メッセージのデータ部分がカプセル化され、対象となるOS32Cと発信元との間でやりとりが完了します。

9.4.11 表20 : SendRRDataコマンド要求

データグラム	フィールド名	フィールド値	データ型	データサイズ	備考
EtherNet/IPヘッダ	Command	0x006F	UINT	2バイト	SendRRDataコマンド
	Length		UINT	2バイト	コマンド固有データの長さ
	Session Handle ID		UDINT	4バイト	Register Session応答メッセージにて受け取ったセッションハンドルのID
	Status	0	UDINT	4バイト	0
	Sender Context Data	0	ARRAY[8]	8バイト	受信側はReserved(保留)フィールドを無視、値は「0」
	Command options	0	UDINT	4バイト	0
コマンド固有データ	Interface Handle ID	0	UDINT	4バイト	発信元（PLCまたはパソコン）が選択したインタフェースハンドル
	Timeout	0	UINT	2バイト	0～65535秒
	Encapsulation packet				セクション9.3および9.4.12 表21 : Get Single Attribute サービスコード要求CPFデータの共通パケットフォーマット仕様を参照。

9.4.12 表21 : Get Single Attributeサービスコード要求CPFデータ

フィールド名	サブフィールド名	フィールド値	データ型	データサイズ	備考
Encapsulation packet	Item count	2	UINT	2バイト	準拠項目数
	Item #1 / Type ID	0	UINT	2バイト	アドレス項目の種類ID
	Item #1 / Length	0	UINT	2バイト	アドレス項目の長さ
	Item #2 / Type ID	0xB2	UINT	2バイト	データ項目の種類ID
	Item #2 / Length	8	UINT	2バイト	データ項目の長さ
	Data Item 1	サービスコード番号	UINT8	1バイト	Get Single AttributeまたはSet Single Attribute要求
	Data Item 2	0x03	UINT8	1バイト	要求パスサイズ（ワード）
	Data Item 3	0x20	UINT8	1バイト	論理クラスの種類
	Data Item 4	クラス番号	UINT8	1バイト	論理クラス番号
	Data Item 5	0x24	UINT8	1バイト	論理インスタンスの種類
	Data Item 6	インスタンス番号	UINT8	1バイト	論理インスタンス番号
	Data Item 7	0x30	UINT8	1バイト	論理属性の種類
	Data Item 8	属性番号	UINT8	1バイト	論理属性番号

9.4.13 表22 : Get Single Attribute要求に対するSendRRDataコマンド応答

データグラム	フィールド名	フィールド値	データ型	データサイズ	備考
EtherNet/IP ヘッダ	Command	0x006F	UINT	2バイト	SendRRDataコマンド
	Length		UINT	2バイト	コマンド固有データの長さ
	Session Handle ID		UDINT	4バイト	Register Session応答メッセージにて受け取ったセッションハンドルのID
	Status	0	UDINT	4バイト	正常に行われた場合のエラー状態は0x0000
	Sender Context Data	Array[8]	オクテット配列	8バイト	要求の値、長さ8
コマンド固有 データ	Command options	0	UDINT	4バイト	
	Interface Handle ID	0	UDINT	4バイト	発信元（PLCまたはパソコン）が選択したインタフェースハンドル
	Timeout	0	UINT	2バイト	0～65535秒
カプセル化パ ケット	Item count	2	UINT	2バイト	準拠項目数
	Item #1 / Type ID	0	UINT	2バイト	アドレス項目の種類ID
	Item #1 / Length	0	UINT	2バイト	アドレス項目の長さ
	Item #2 / Type ID	0xB2	UINT	2バイト	データ項目の種類ID
	Item #2 / Length	8 + データの長さ	UINT	2バイト	データ項目の長さ
	Data Item 1	応答	UINT8	1バイト	Get Single AttributeまたはSet Single Attribute応答コード
	Reserved field	0	UINT8	1バイト	Reserved(保留)フィールドは使用しない
	Data Item 2	Status	UINT8	1バイト	Get Single AttributeまたはSet Single Attributeエラー状態コード
	Data Item 3	追加状態サイズ	UINT8	1バイト	追加のエラー状態情報（16ビットWord）の数
属性データ	Data Array		ARRAY[]	データの長さ	属性データがある場合

9.4.14 表23 : Set Single Attributeサービスコード要求CPFデータ

フィールド名	サブフィールド名	フィールド値	データ型	データサイズ	備考
Encapsulation packet	Item count	2	UINT	2バイト	準拠項目数
	Item #1 / Type ID	0	UINT	2バイト	アドレス項目の種類ID
	Item #1 / Length	0	UINT	2バイト	アドレス項目の長さ
	Item #2 / Type ID	0xB2	UINT	2バイト	データ項目の種類ID
	Item #2 / Length	8 + データの長さ	UINT	2バイト	データ項目の長さ
	Data Item 1	サービスコード番号	UINT8	1バイト	Get Single AttributeまたはSet Single Attribute要求
	Data Item 2	0x03	UINT8	1バイト	要求パスサイズ（ワード）
	Data Item 3	0x20	UINT8	1バイト	論理クラスの種類
	Data Item 4	クラス番号	UINT8	1バイト	論理クラス番号
	Data Item 5	0x24	UINT8	1バイト	論理インスタンスの種類
	Data Item 6	インスタンス番号	UINT8	1バイト	論理インスタンス番号
	Data Item 7	0x30	UINT8	1バイト	論理属性の種類
	Data Item 8	属性番号	UINT8	1バイト	論理属性番号
	Data Array		ARRAY[]	属性データの長さ	属性データ

9.4.15 表24 : Set Single Attribute要求に対するSendRRDataコマンド応答

データグラム	フィールド名	フィールド値	データ型	データサイズ	備考
EtherNet/IP ヘッダ	Command	0x006F	UINT	2バイト	SendRRDataコマンド
	Length		UINT	2バイト	コマンド固有データの長さ
	Session Handle ID		UDINT	4バイト	Register Session応答メッセージにて受け取ったセッションハンドルのID
	Status	0	UDINT	4バイト	正常に行われた場合のエラー状態は0x0000
	Sender Context Data	Array[8]	オクテット配列	8バイト	要求の値、長さ8
	Command options	0	UDINT	4バイト	
コマンド固有データ	Interface Handle ID		UDINT	4バイト	発信元（PLCまたはパソコン）が選択したインタフェースハンドル
	Timeout	0	UINT	2バイト	0～65535秒
カプセル化パケット	Item count	2	UINT	2バイト	準拠項目数
	Item #1 / Type ID	0	UINT	2バイト	アドレス項目の種類ID
	Item #1 / Length	0	UINT	2バイト	アドレス項目の長さ
	Item #2 / Type ID	0xB2	UINT	2バイト	データ項目の種類ID
	Item #2 / Length	8 + データの長さ	UINT	2バイト	データ項目の長さ
	Data Item 1	応答	UINT8	1バイト	Get Single AttributeまたはSet Single Attribute応答コード
	Reserved field	0	UINT8	1バイト	Reserved(保留)フィールドは使用しない
	Data Item 2	Status	UINT8	1バイト	Get Single AttributeまたはSet Single Attributeエラー状態コード
	Data Item 3	追加状態サイズ	UINT8	1バイト	追加のエラー状態情報（16ビットWord）の数

SendRRData コマンドは、発信元（パソコンまたはPLC）と対象デバイス（OS32C）との間にUDP I/O 接続を確立する場合にも使われます。セクション9.4.16 表25 : Large Forward Open要求のカプセル化パケットでは、OS32CからUDP I/O接続によるストリーミングを開始する際に、SendRRData コマンドとともに使用する共通パケットフォーマットを記載しています。

9.4.16 表25 : Large Forward Open要求のカプセル化パケット

データグラム	フィールド名	フィールド値	データ型	データサイズ	備考
EtherNet/IP ヘッダ	Command	0x006F	UINT	2バイト	SendRRData コマンド
	Length		UINT	2バイト	コマンド固有データの長さ
	Session Handle ID		UDINT	4バイト	Register Session 応答メッセージにて受け取ったセッションハンドルのID
	Status	0	UDINT	4バイト	正常に行われた場合のエラー状態は0x0000
	Sender Context Data	Array[8]	オクテット配列	8バイト	要求の値、長さ8
コマンド固有 データ	Command options	0	UDINT	4バイト	
	Interface Handle ID		UDINT	4バイト	発信元（PLCまたはパソコン）が選択したインタフェースハンドル
	Timeout	0	UINT	2バイト	0～65535秒
カプセル化 パケット	Item count	2	UINT	2バイト	準拠項目数
	Item #1 / Type ID	0	UINT	2バイト	アドレス項目の種類ID
	Item #1 / Length	0	UINT	2バイト	アドレス項目の長さ
	Item #2 / Type ID	0xB2	UINT	2バイト	データ項目の種類ID
	Item #2 / Length	(ex.54)	UINT	2バイト	データ項目の長さ、下記に続くデータの長さ
	Data Item 1	0x5B	UINT8	1バイト	Large Forward Open 要求
	Data Item 2	0x02	UINT8	1バイト	要求バスサイズ（ワード）
	Data Item 3	0x20	UINT8	1バイト	論理クラスの種類
	Data Item 4	0x06	UINT8	1バイト	論理クラス要求
	Data Item 5	0x24	UINT8	1バイト	論理インスタンスの種類
	Data Item 6	0x01	UINT8	1バイト	論理インスタンス要求
	Data Item 7	0x06	UINT8	1バイト	優先タイムティック
	Data Item 8	(ex.880)	UINT8	1バイト	タイムアウトティック（ms）
	Data Item 9	0	UDINT	4バイト	発信元→対象識別番号
	Data Item 10	任意	UDINT	4バイト	対象→発信元識別番号、ここではランダム数を使用
	Data Item 11	任意	UINT	2バイト	接続シリアルナンバー
	Data Item 12	405	UINT	2バイト	ベンダー識別番号、ここではORTベンダー IDを使用可能
	Data Item 13	任意	UDINT	4バイト	発信元シリアルナンバー
	Data Item 14	0	UINT8	1バイト	接続タイムアウト乗数（デフォルト = 0）
	Data Item 15	0	UINT8	1バイト	保留
	Data Item 16	0	UINT8	1バイト	保留
	Data Item 17	0	UINT8	1バイト	保留
	Data Item 18	任意 (例 : 880000)	UDINT	4バイト	発信元→対象繰り返しパケット間隔（RPI）（ミリ秒 : us）
	Data Item 19	(例 : 0x48000008)	UDINT	4バイト	発信元→対象接続パラメータ EDSファイル参照
	Data Item 20	任意 (例 : 80000)	UDINT	4バイト	対象→発信元繰り返しパケット間隔（RPI）（ミリ秒 : us）

データグラム	フィールド名	フィールド値	データ型	データサイズ	備考
	Data Item 21	(例 : 0x48000588)	UDINT	4バイト	対象→発信元接続パラメータ EDSファイル参照
	Data Item 22	0x01	UINT8	1バイト	トランスポートクラストリガ、クライアント
	Data Item 23	0x04	UINT8	1バイト	接続バスのワード数
	Data Item 24	0x20	UINT8	1バイト	接続バスクラス、インスタンス
	Data Item 25	0x04	UINT8	1バイト	アセンブリオブジェクト接続バス論理クラス
	Data Item 26	0x24	UINT8	1バイト	接続バス論理インスタンスセグメント
	Data Item 27	0x01	UINT8	1バイト	接続バス論理インスタンス
	Data Item 28	0x2c	UINT8	1バイト	接続ポイント、発信元→対象
	Data Item 29	任意(例:112)	UINT8	1バイト	出力アセンブリ番号 EDSファイル参照
	Data Item 30	0x2c	UINT8	1バイト	接続ポイント、対象→発信元
	Data Item 31	任意(例:102)	UINT8	1バイト	入力アセンブリ番号 EDSファイル参照

9.4.17 表26 : Large Forward Open要求に対するSendRRDataコマンド応答

データグラム	フィールド名	フィールド値	データ型	データサイズ	備考
EtherNet/IP ヘッダ	Command	0x006F	UINT	2バイト	SendRRDataコマンド
	Length		UINT	2バイト	コマンド固有データの長さ
	Session Handle ID		UDINT	4バイト	Register Session応答メッセージにて受け 取ったセッションハンドルのID
	Status	0	UDINT	4バイト	正常の場合「0」
	Sender Context Data	Array[8]	オクテット配列	8バイト	要求の値、長さ8
	Command options	0	UDINT	4バイト	
コマンド固有 データ	Connection Interface Handle		UDINT	4バイト	データストリームで使用する接続インタ フェースハンドル
	Timeout Period		UINT	2バイト	タイマで使用する接続タイムアウト期間
カプセル化 データ	Item count	4	UINT	2バイト	準拠項目数
	Item #1 / Type ID	0	UINT	2バイト	アドレス項目の種類ID
	Item #1 / Length	0	UINT	2バイト	アドレス項目の長さ
	Item #2 / Type ID	0xB2	UINT	2バイト	データ項目の種類ID
	Item #2 / Length	データ長	UINT	2バイト	データ項目の長さ、下記に続くデータの 長さ
	Data Item 1	応答 (例:0xDB)	UINT8	1バイト	応答コード
	Reserved field	0	UINT8	1バイト	Reserved(保留)フィールドは使用しない
	Data Item 2	Status	UINT8	1バイト	状態コード
	Data Item 3	追加状態 サイズ	UINT8	1バイト	追加状態サイズデータ。「0」でない場合 は、記載のエラー状態記述を確認。
	Data Item 4		UDINT	4バイト	発信元→対象識別番号 (ID)
	Data Item 5		UDINT	4バイト	対象→発信元識別番号 (ID)
	Data Item 6		UINT	2バイト	接続シリアルナンバー
	Data Item 7		UINT	2バイト	発信元デバイスベンダー ID番号
	Data Item 8		UDINT	4バイト	発信元デバイスシリアルナンバー
	Data Item 9		UDINT	4バイト	発信元→対象承認パケット間隔 (API)
	Data Item 10		UDINT	4バイト	対象→発信元承認パケット間隔 (API)
	Data Item 11	データサイズ	UINT8	1バイト	アプリケーションデータサイズ
	Data Array		ARRAY□	データサイズ の長さ	データがある場合

UDP I/O 接続が確立されると、対象デバイスは、承認されたパケット間隔でアセンブリデータを発信元デバイスにストリーミングします。セクション9.4.18 表27：UDP I/O 接続パケット（下記）では、OS32C（対象デバイス）によって送られたデータストリームのフォーマットを記載しています。

9.4.18 表27：UDP I/O接続パケット

フィールド名	サブフィールド名	フィールド値	データ型	データサイズ	備考
Encapsulation packet	Item count		UINT	2バイト	準拠項目数
	Item #1 / Type ID		UINT	2バイト	アドレス項目の種類ID
	Item #1 / Length		UINT	2バイト	アドレス項目の長さ
	Item #1 / Connection ID		UDINT	4バイト	接続ID番号
	Item #1 / Sequence ID		UDINT	4バイト	連続ID番号
	Item #2 / Type ID		UINT	2バイト	データ項目の種類ID
	Item #2 / Length		UINT	2バイト	データ項目の長さ、下記に続くデータの長さ
	Sequence Number		UINT	2バイト	パケット連続番号
	Data Array		ARRAY□	データの長さ、長さ：2バイト	アセンブリオブジェクトデータ

UDP I/O接続を終了するには、発信元デバイスがForward Close要求とともにSendRRDataコマンドを発信する必要があります。セクション9.4.19 表 28 : Forward Close 要求のカプセル化パケット（下記）では、OS32C（対象デバイス）に発信するカプセル化メッセージのフォーマットを記載しています。

9.4.19 表28 : Forward Close要求のカプセル化パケット

データグラム	フィールド名	フィールド値	データ型	データサイズ	備考
EtherNet/IP ヘッダ	Command	0x006F	UINT	2バイト	SendRRDataコマンド
	Length		UINT	2バイト	コマンド固有データの長さ
	Session Handle ID		UDINT	4バイト	Register Session応答メッセージにて受け取ったセッションハンドルのID
	Status	0	UDINT	4バイト	
	Sender Context Data	Array[8]	オクテット配列	8バイト	要求の値、長さ8
	Command options	0	UDINT	4バイト	
コマンド固有 データ	Connection Interface Handle		UDINT	4バイト	データストリームで使用する接続インタフェースハンドル
	Timeout Period		UINT	2バイト	タイマで使用する接続タイムアウト 期間
カプセル化 パケット	Item count	2	UINT	2バイト	準拠項目数
	Item #1 / Type ID	0	UINT	2バイト	アドレス項目の種類ID
	Item #1 / Length	0	UINT	2バイト	アドレス項目の長さ
	Item #2 / Type ID	0xB2	UINT	2バイト	データ項目の種類ID
	Item #2 / Length	(例:0x16)	UINT	2バイト	データ項目の長さ、下記に続くデータの長さ
	Data Item 1	0x4E	UINT8	1バイト	Forward Close要求
	Data Item 2	0x02	UINT8	1バイト	バスにおける16ビットワードの数
	Data Item 3	0x20	UINT8	1バイト	論理クラスセグメント
	Data Item 4	0x06	UINT8	1バイト	論理クラス要求 メッセージルーター
	Data Item 5	0x24	UINT8	1バイト	論理インスタンスセグメント
	Data Item 6	0x01	UINT8	1バイト	論理インスタンス要求
	Data Item 7	0x06	UINT8	1バイト	優先タイムティック
	Data Item 8	(例:880)	UINT8	1バイト	タイムアウトティック (ms)
	Data Item 9		UDINT	4バイト	発信元シリアルナンバー
	Data Item 10	2	UINT8	1バイト	バスにおける16ビットワードの数
	Data Item 11	0	UINT8	1バイト	保留
	Data Item 12	0x20	UINT8	1バイト	論理クラスセグメント
	Data Item 13	0x02	UINT8	1バイト	論理クラス要求 接続マネージャ
	Data Item 14	0x24	UINT8	1バイト	論理インスタンスセグメント
	Data Item 15	0x01	UINT8	1バイト	論理インスタンス要求

9.4.20 表29 : Forward Close要求に対するSendRRDataコマンド応答

データグラム	フィールド名	フィールド値	データ型	データサイズ	備考
EtherNet/IP ヘッダ	Command	0x006F	UINT	2バイト	SendRRDataコマンド
	Length		UINT	2バイト	コマンド固有データの長さ
	Session Handle ID		UDINT	4バイト	Register Session応答メッセージにて受け取ったセッションハンドルのID
	Status	0	UDINT	4バイト	正常の場合「0」
	Sender Context Data	Array[8]	オクテット配列	8バイト	要求の値、長さ8
コマンド固有 データ	Command options	0	UDINT	4バイト	
	Connection Interface Handle		UDINT	4バイト	データストリームで使用する接続インタフェースハンドル
	Timeout Period		UINT	2バイト	タイマで使用する接続タイムアウト期間
カプセル化 データ	Item count	2	UINT	2バイト	準拠項目数
	Item #1 / Type ID	0	UINT	2バイト	アドレス項目の種類ID
	Item #1 / Length	0	UINT	2バイト	アドレス項目の長さ
	Item #2 / Type ID	0xB2	UINT	2バイト	データ項目の種類ID
	Item #2 / Length	データ長	UINT	2バイト	データ項目の長さ、下記に続くデータの長さ
	Data Item 1	応答 (例:0xCE)	UINT8	1バイト	応答コード
	Reserved field	0	UINT8	1バイト	Reserved(保留)フィールドは使用しない
	Data Item 2	Status	UINT8	1バイト	状態コード
	Data Item 4		UINT	2バイト	接続シリアルナンバー
	Data Item 5		UINT	2バイト	発信元デバイスベンダー ID番号
	Data Item 6		UDINT	4バイト	発信元デバイスシリアルナンバー
	Data Item 7	データサイズ	UINT8	1バイト	アプリケーションデータサイズ アプリケーションデータ配列における16ビットワードの数
	Reserved field	0	UINT8	1バイト	Reserved(保留)フィールドは使用しない
	Data Array		ARRAY[]	データサイズの長さ	アプリケーションデータがある場合

10. アプリケーション例

10.1 Explicit TCP/IP要求・返信メッセージを使ったランタイムモニタリング

10.1.1 ネットワーク設定

OS32C設定ツールを使用して、OS32CのIPアドレス、サブネットマスク、デフォルトゲートウェイを設定します（OS32Cユーザーズマニュアルを参照してください）。

10.1.2 パソコン・PLCの設定とプロセス制御の例

OS32CのEDSファイルに記載されているexplicitメッセージコマンド情報を使ってパソコンまたはPLCを設定し、OS32CのEtherNet/IP Explicitメッセージ機能にアクセスし、計測レポートデータをモニタリングできるようにします。ここでは、OS32Cのランタイムモニタリング属性を -0.4° から 270.4° の範囲内で2か所のセクターをモニタリングします。それに必要な設定とTCP/IP要求・応答コマンドは、下記のようになります。

10.1.2.1 OS32Cの防護エリア、警告エリアを設定する

1. OS32C設定ツールを使って、OS32Cの防護エリアの範囲が500 mm以内になるように設定します。
2. 上記ツールを使用して、警告エリア1をセクター1（ $0\sim 45^{\circ}$ ）における1000 mmの検知エリア範囲に設定します。
3. 上記ツールを使用して、警告エリア2をセクター2（ $225\sim 270^{\circ}$ ）における1500 mmの検知エリア範囲に設定します。

10.1.2.2 ソフトウェアドライバの通信ポートを設定する

- ・UDP/IPポート44818を初期化して、接続されているすべてのOS32Cに対して認識が行える状態にします。List identityコマンドを使うと、どのデバイスがOS32Cであるかを特定することができます。
- ・TCP/IPポート44818を初期化して、explicitメッセージをすべて受信・送信できるようにし、OS32Cとの接続を確立します。
- ・UDP I/Oが必要な場合は、UDP/IPポート2222を初期化して、I/O接続によるメッセージをすべて送受信できるようにします。

10.1.2.3 既知のIPアドレスを使ってOS32CとのTCP接続を確立する

適切なソケットプログラム言語を使用して、パソコンのローカルIPアドレスとOS32CのIPアドレスによるTCP/IP接続を確立します。

10.1.2.4 OS32Cを識別する

セクション9.4.1のList Identityコマンドを使って、OS32Cのシリアルナンバーと製品名を特定します。

パソコン → OS32C	
データグラム構造	セクション9.4.2 表14 : List Identityコマンド要求を参照
サイズ	24バイト
更新パラメータ	EIP_UINT16 Timeout:0x01F4 = 500ミリ秒
16進法	63 00 00 00 00 00 00 00 00 00 00 00 00 00 <u>F4 01</u> 00 00 00 00 00 00 00 00 00

OS32C → パソコン	
データグラム構造	セクション9.4.3 表15 : List Identityコマンド応答を参照
サイズ	90バイト
List Identityコマンド応答例	
返される 追加パラメータ	EIP_UINT16 item_count = 0x0001 EIP_UINT16 item_id = 0x000C EIP_UINT16 item_length = 0x003C EIP_UINT16 encap_version = 0x0001 EIP_UINT8 sock_family[2] = 0x0002 (ビッグエンディアン) EIP_UINT8 sock_port[2] = 0x12AF (ビッグエンディアン) EIP_UINT8 sock_addr[4] = 0x0101A8C0 (ビッグエンディアン) EIP_UINT8 sock_zero[8] = 00 00 00 00 00 00 00 00 (ビッグエンディアン) EIP_UINT16 vendor_id = 0x0195 (405) EIP_UINT16 device_type = 0x002B (ジェネリック) EIP_UINT16 product_code = 0x000C (12, OS32C) EIP_UINT8 revision[2] = 0x030B (v3.11) EIP_UINT16 current_status = 0x0000 EIP_UINT16 serial_number_low = 0x0649 (AS01609) EIP_UINT16 serial_number_high = 0x0000 EIP_UINT8 name_length = 0x1A (26) EIP_UINT8 product_name[name_length] = 4F 53 33 32 43 20 53 61 66 65 74 79 20 4C 61 73 65 72 20 53 63 61 6E 6E 65 72 (OS32Cセーフティレーザスキャナ) EIP_UINT8 product_state = 0x03 (操作可能)
16進法	63 00 42 00 00 00 00 00 00 00 00 00 00 <u>F4 01</u> 00 00 00 00 00 00 00 00 00 01 00 <u>0C 00 3C 00 01 00 00 02 AF 12 C0 A8 01 01</u> 00 00 00 00 00 00 00 00 <u>00 95 01 2B 00 0C 00 03 0B 00 00 49 06 00 00 1A 4F 53 33 32 43 20 53</u> <u>61 66 65 74 79 20 4C 61 73 65 72 20 53 63 61 6E 6E 65 72 03</u>

10.1.2.6 OS32Cの計測距離フォーマットを設定する

EtherNet/IP SendRRData コマンドを使って、OS32C の検出距離計測フォーマットを設定するための explicit TCP/IP メッセージを送信します。パラメータは下記のとおりです。

- o Service code 16 (0x10) // 単一属性設定
- o Object class 115 (0x73) // ベンダー固有オブジェクトのクラス
- o Instance 1 (0x01) // ベンダー固有インスタンス
- o Attribute 4 (0x04) // 検出距離計測レポートフォーマット
- o UINT16 Data // Data = 1, RANGE_MEASURE_50M (デフォルト値)

パソコン → OS32C	
データグラム構造	セクション9.4.11 表20 : SendRRData コマンド 要求を参照 セクション9.4.14 表23 : Set Single Attribute サービスコード 要求 CPF データを参照
サイズ	50 バイト
更新パラメータ	EIP_UINT16 length = 0x001A (コマンド固有データの長さ) EIP_UINT32 session_handle_id = 0x00000005 EIP_UINT8 service_code = 0x10 (Set_Single_Attribute) EIP_UINT8 class_code = 0x73 (ベンダーオブジェクト、計測設定) EIP_UINT8 instance_id = 0x01 (ベンダーオブジェクトインスタンス) EIP_UINT8 attribute_id = 0x04 (検出距離フォーマット属性番号) EIP_UINT16 attribute data = 0x0001 (検出距離フォーマット設定)
16進法	6F 00 1A 00 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 02 00 00 00 00 00 B2 00 0A 00 10 03 20 73 24 01 30 04 01 00

OS32C → パソコン	
データグラム構造	セクション9.4.15 表24 : Set Single Attribute 要求に対する SendRRData コマンド 応答を参照
サイズ	44 バイト
SendRRData コマンド 応答例	
返される更新パラメータ	EIP_UINT16 length = 0x0014 (コマンド固有データの長さ) EIP_UINT16 data_length = 0x0004 (サービス応答データの4バイト) EIP_UINT8 service_response = 0x90 (Set_Single_Attribute 0x80) EIP_UINT8 service_status = 0x00 (正常)
16進法	6F 00 14 00 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 02 00 00 00 00 00 B2 00 04 00 90 00 00 00 00

10.1.2.8 OS32Cの計測ビーム選択マスクを設定する

EtherNet/IP SendRRDataコマンドを使って、OS32Cのビーム選択マスクを設定するためのexplicit TCP/IPメッセージを送信します。パラメータは下記のとおりです。

- o Service code 16 (0x10) // 単一属性設定
- o Object class 115 (0x73) // ベンダー固有オブジェクトのクラス
- o Instance 1 (0x01) // ベンダー固有インスタンス
- o Attribute 12 (0x0C) // ビーム選択マスク
- o UINT16 Data [44] // 必要なデータは、下記のアルゴリズムを使って選択可能

```
// ビーム選択マスクを定義
memset((uint8 *)&BeamReportMask, ZERO, sizeof(BeamReportMask));
SetupBeamSelectionMask(0, 45, BeamReportMask);
SetupBeamSelectionMask(225, 270, BeamReportMask);
/*****
** 関数・タスク: SetupBeamSelectionMask
** 目的: 計測データレポート中で使うビームを選択
** 引数: float startAngle (-0.4 to 270.4 degrees)
**       float endAngle (-0.4 to 270.4 degrees)
**       uint16 *BeamSelectionData (Beam Selection Array へのポインタ)
** 戻り値: bool TRUE/FALSE (PASS/FAIL)
*****/
bool SetupBeamSelectionMask(float startAngle, float endAngle, uint16 *BeamSelectionData)
{
    int16 i;
    int16 startBeam = ((startAngle * 10) / 4); // startAngle/ 各ビーム 0.4°
    int16 endBeam = (((endAngle * 10) + 2) / 4); // endAngle/ 各ビーム 0.4°

    uint32 beamSelectionMask = 0x0001;
    uint32 beamSelctionCtr = 0;

    if ((startBeam < -1) || (startBeam > 677) || (endBeam < -1) || (endBeam > 677) || (startBeam > endBeam)) {
        return(FALSE);
    }
    // setup initial mask variables.
    for (i = -1; i < startBeam; i++){
        if (beamSelectionMask == 0x8000) {
            beamSelctionCtr++;
            beamSelectionMask = 0x0001;
        } else {
            beamSelectionMask = beamSelectionMask << 1;
        } // if else の終了
    } // for の終了
    // 必要な計測ビームのセットアップ
    for (i = startBeam; i < endBeam; i++) {
        BeamSelectionData[beamSelctionCtr] |= beamSelectionMask;
        if (beamSelectionMask == 0x8000) {
            beamSelctionCtr++;
            beamSelectionMask = 0x0001;
        } else {
            beamSelectionMask = beamSelectionMask << 1;
        } // if else の終了
    } // for の終了
    return (TRUE);
} // SetupBeamSelectionMask() の終了
```

OS32C → パソコン	
データグラム構造	セクション9.4.15 表24：Set Single Attribute要求に対するSendRRDataコマンド応答を参照
サイズ	44バイト
SendRRDataコマンド 応答例	
返される 更新パラメータ	EIP_UINT16 length = 0x0014 (コマンド固有データの長さ) EIP_UINT16 data_length = 0x0004 (サービス応答データの4バイト) EIP_UINT8 service_response = 0x90 (Set_Single_Attribute 0x80) EIP_UINT8 service_status = 0x00 (正常)
16進法	6F 00 14 00 <u>05 00 00 00</u> 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 02 00 00 00 00 00 B2 00 <u>04 00 90</u> 00 <u>00 00</u>

1. OS32Cの計測データを読み込むには、SendRRDataコマンドを用いてexplicit TCP/IPメッセージを送信し、対象となるアセンブリオブジェクト全体のデータを読み込みます。入力アセンブリ102および103からのデータを要求した場合、OS32Cから返されるデータはスキャン周期と同期していません。スキャン周期と同期したスキャンデータを読み込むには、ベンダー固有アセンブリ114、116、および117を使用する必要があります。ベンダー固有オブジェクトのデータ要求に対して、OS32Cは、現在のスキャンが終了し、次のスキャンが始まる前のデータを返します。TCP要求・応答explicitメッセージでは、サービスコード14 (0x0E、Get_Single_Attribute) を使用し、以下のパラメータを設定します。

```
サービスコード 14 (0x0E) // 単一属性取得
Object class 0x04 (0x04) // アセンブリオブジェクトクラス
Instance number // アセンブリ番号、例：102 (0x66)
Attribute 03 (0x03) // アセンブリデータ
```

```
サービスコード 14 (0x0E) // 単一属性取得
Object class number // ベンダー固有オブジェクト、例 : 114 (0x72)
Instance 1 (0x01) // インスタンス番号
Attribute 03 (0x03) // アセンブリデータ
```

パソコン→OS32C	
データグラム構造	セクション9.4.11 表20 : SendRRDataコマンド要求を参照 セクション9.4.12 表21 : Get Single Attributeサービスコード要求CPFデータを参照
サイズ	48バイト
更新パラメータ	EIP_UINT16 length = 0x0018 (コマンド固有データの長さ) EIP_UINT32 session_handle_id = 0x00000005 EIP_UINT8 service_code = 0x0E (Get_Single_Attribute) EIP_UINT8 class_code = 0x72 (ベンダーオブジェクト114、検出距離計測データ) EIP_UINT8 instance_id = 0x01 (ベンダーオブジェクトインスタンス) EIP_UINT8 attribute_id = 0x03 (アセンブリオブジェクトデータ)
16進法	6F 00 18 00 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 02 00 00 00 00 00 B2 00 08 00 <u>0E</u> 03 20 <u>72</u> 24 <u>01</u> 30 <u>03</u>

OS32C →パソコン	
データグラム構造	セクション9.4.13 表22 : Get Single Attribute要求に対するSendRRDataコマンド応答を参照
サイズ	140バイト (例 : 計測レポートのビーム数が20)
SendRRDataコマンド応答例	
返される更新パラメータ	EIP_UINT16 length = 0x0074 (コマンド固有データの長さ) EIP_UINT16 data_length = 0x0064 (100バイト = サービス応答4バイト + 計測レポートヘッダ56バイト + ビームデータ40バイト) EIP_UINT8 service_response = 0x8E (Get_Single_Attribute 0x80) EIP_UINT8 service_status = 0x00 (正常) EIP_UINT8 Data_Array[data_length - 4]; セクション4.8 表8 : 共通計測レポートヘッダフォーマット (56バイト) を参照 注意 : ビーム数 = 0x0014 (20) セクション4.9 表9 : 入力アセンブリ102およびベンダー固有オブジェクト114 (最大1410バイト) を参照
16進法	6F 00 74 00 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 02 00 00 00 00 B2 00 <u>64</u> 00 8E 00 00 00 <u>D3 D7 02</u> 00 38 98 00 00 28 7E C4 B0 21 A9 00 00 03 00 07 00 00 00 00 00 00 00 00 00 00 00 01 00 00 00 02 01 88 33 AE 31 00 00 00 00 00 00 00 00 00 00 00 00 01 <u>00 01 00 00 00 14 00</u> 4D 08 3E 08 4E 08 3D 08 4E 08 53 08 4E 08 3D 08 5B 08 53 08 5B 08 53 08 5B 08 61 08 65 08 61 08 65 08 63 08 65 08 6D 08

- OS32Cとの常時接続を維持するには、OS32Cからのデータを受信してから2ミリ秒の遅延後、アプリケーションコードによってTCP/IP要求を再送信する必要があります。OS32Cは次のスキャンが終了した時点で新しいデータを返します。
- 接続を終了するには、アプリケーションコードによって、まずセクション 9.4.8 に記載されている UnRegister Sessionコマンドを送信し、次に該当するTCP/IPソケットを閉じます。

パソコン→OS32C	
データグラム構造	セクション9.4.9 表19 : Un-Register Sessionコマンド要求を参照
サイズ	24バイト
更新パラメータ	EIP_UINT32 session_handle_id = 0x00000005
16進法	66 00 00 00 <u>05 00 00 00</u> 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

OS32C →パソコン	
注意	このコマンドに対する応答はありません。 このコマンドを受け取った時点で、OS32CはTCP/IPソケット接続を閉じます。
サイズ	0バイト
Un-Register Session応答例	
16進法	なし

10.2 Implicit UDP I/O接続を使ったランタイムモニタリング

10.2.1 ネットワーク設定

OS32C設定ツールを使用して、OS32CのIPアドレス、サブネットマスク、デフォルトゲートウェイを設定します（OS32Cユーザズマニュアルを参照してください）。

10.2.2 パソコン・PLCの設定とプロセス制御の例

本書に掲載されているI/Oメッセージコマンド情報を使ってパソコンまたはPLCを設定し、OS32CのI/Oメッセージ機能にアクセスし、計測レポートデータをモニタリングできるようにします。ここでは、OS32Cのランタイムモニタリング属性を0から270°の範囲内で2か所のセクターをモニタリングします。それに必要な設定とTCP/IP要求・応答コマンドは、下記のようになります。

10.2.2.1 OS32Cの防護エリア、警告エリアを設定する

1. OS32C設定ツールを使って、OS32Cの防護エリアの範囲が500 mm以内になるように設定します。
2. 上記ツールを使用して、警告エリア1をセクター1（0～45°）における1000 mmの検知エリア範囲に設定します。
3. 上記ツールを使用して、警告エリア2をセクター2（225～270°）における1500 mmの検知エリア範囲に設定します。

10.2.2.2 ソフトウェアドライバの通信ポートを設定する

- ・UDP/IPポート44818を初期化して、接続されているすべてのOS32Cに対して認識が行える状態にします。List identityコマンドを使うと、どのデバイスがOS32Cであるかを特定することができます。
- ・TCP/IPポート44818を初期化して、explicitメッセージをすべて受信・送信できるようにし、OS32Cの接続を確立します。
- ・アプリケーションにUDP I/Oが必要な場合は、UDP/IPポート2222を初期化して、I/O接続によるメッセージをすべて送受信できるようにします。

10.2.2.3 既知のIPアドレスを使ってOS32CとのTCP接続を確立する

適切なソケットプログラム言語を使用して、パソコンのローカルIPアドレスとOS32CのIPアドレスによるTCP/IP接続を確立します。

10.2.2.4 OS32Cとの通信セッションを登録する

TCP/IP例で記述したセクション10.1.2.5 OS32Cとの通信セッションを登録するを参照。

10.2.2.5 OS32Cの計測距離フォーマットを設定する

TCP/IP例で記述したセクション10.1.2.6 OS32Cの計測距離フォーマットを設定するを参照。

10.2.2.6 OS32Cの計測反射率フォーマットを設定する

TCP/IP例で記述したセクション10.1.2.7 OS32Cの計測反射率フォーマットを設定するを参照。

10.2.2.7 OS32Cの計測ビーム選択マスクを設定する

TCP/IP例で記述したセクション10.1.2.8 OS32Cの計測ビーム選択マスクを設定するを参照。

10.2.2.8 発信元（パソコン）とOS32C間にLarge Forward Open I/O接続を作成する

セクション9.4.10に記載のSendRRDataコマンドとセクション9.4.16のLarge Forward Open要求を用いて、OS32CとのI/O接続を作成し、タイムアウト期間800ミリ秒、80ミリ秒間隔でアセンブリオブジェクト102をストリーミングします。

パソコン→OS32C	
データグラム構造	セクション9.4.11 表20：SendRRDataコマンド要求を参照 セクション9.4.16 表25：Large Forward Open要求のカプセル化パケットを参照
サイズ	94バイト
更新パラメータ	EIP_UINT16 length = 0x0046 (コマンド固有データの長さ) EIP_UINT32 session_handle_id = 0x00000005 EIP_UINT32 interface_handle_id = 0x00000000 EIP_UINT16 timeout = 0x0050 (80ミ リ秒) EIP_UINT16 data_item_length = 0x0036 (54バイト) EIP_UINT8 service request = 0x5B (Large Forward Open) EIP_UINT16 timeout_ticks = 0x0050 (80ミ リ秒) EIP_UINT32 o_to_t_connection_id = 0x00020003 (ユーザー選択可) EIP_UINT32 t_to_o_connection_id = 0x00020004 (ユーザー選択可) EIP_UINT16 connection_serial_number = 0x6789 (ユーザー選択可) EIP_UINT16 originator_vendor_id = 0x0195 (405, ORT) EIP_UINT32 originator_serial_number = 0x00004321 (ユーザー選択可) EIP_UINT32 o_to_t_rpi = 0x00177FA0 (発信元から対象への繰り返しパケット間隔) EIP_UINT32 o_to_t_connection_parameters = 0x48000000 + 6E (出力アセンブリサイズ + 6) EIP_UINT32 t_to_o_rpi = 0x00013070 (対象から発信元への繰り返しパケット間隔) EIP_UINT32 t_to_o_connection_parameters = 0x4A000000 + 0x0584 (入力アセンブリサイズ + 2) EIP_UINT8 output_assembly_number = 0x71 (出力アセンブリ113をトリガとして使用) EIP_UINT8 input_assembly_number = 0x66 (入力アセンブリ102)
16進法	<u>6F 00 46 00 05 00 00 00</u> 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 <u>00 00 00 00 50 00 02 00 00 00 00 00 B2 00 36 00</u> 5B 02 20 06 24 01 06 <u>50 03 00 02 00 04 00 02 00 89 67 95 01 21 43 00 00</u> 00 00 00 A0 7E <u>17 00 6E 00 00 48 70 30 01 00 84 05 00 4A</u> 01 04 20 04 24 01 2C 71 2C 66

OS32C → パソコン	
データグラム構造	セクション9.4.17 表26 : Large Forward Open要求に対するSendRRDataコマンド応答を参照
サイズ	110バイト
Large Forward Openに対するSendRRData応答例	
返される 更新パラメータ	EIP_UINT16 length = 0x0056 (コマンド固有データの長さ) EIP_UINT32 session_handle_id = 0x00000005 EIP_UINT16 data_item_length = 0x001E (30バイト) EIP_UINT8 service_response = 0xDB (Large Forward Open 0x80) EIP_UINT32 o_to_t_connection_id = 0x780C0002 (OS32Cが選択したid) EIP_UINT32 t_to_o_connection_id = 0x00020004 (OS32Cが承認したid) EIP_UINT16 connection_serial_number = 0x6789 EIP_UINT16 originator_vendor_id = 0x0195 (405, ORT) EIP_UINT32 originator_serial_number = 0x00004321 EIP_UINT32 o_to_t_rpi = 0x001781D0 (発信元から対象への実際のパケット間隔) EIP_UINT32 t_to_o_rpi = 0x00013070 (対象から発信元への実際のパケット間隔) EIP_UINT8 application_reply_size = 0x00 (16ビットワードの数) EIP_UINT8 reserved = 0x00 注: 残りのデータは使用されません。
16進法	6F 00 <u>56</u> <u>00</u> <u>05</u> <u>00</u> <u>00</u> <u>00</u> 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 04 00 00 00 00 00 B2 00 <u>1E</u> <u>00</u> <u>DB</u> 00 00 00 <u>02</u> <u>00</u> <u>0C</u> <u>78</u> <u>04</u> <u>00</u> <u>02</u> <u>00</u> <u>89</u> <u>67</u> <u>95</u> <u>01</u> <u>21</u> <u>43</u> <u>00</u> <u>00</u> <u>D0</u> <u>81</u> <u>17</u> <u>00</u> <u>D8</u> <u>5C</u> <u>02</u> <u>00</u> <u>00</u> <u>00</u> 00 80 10 00 00 02 08 AE C0 A8 01 01 00 00 00 00 00 00 00 00 01 80 10 00 00 02 08 AE C0 A8 01 07 00 00 00 00 00 00 00 00

10.2.2.9 発信元（パソコン）とOS32C間のI/O接続をキープアライブする

UDP I/O出力アセンブリによるトリガを送り、I/O接続をキープアライブし、新しいデータがあれば更新するようにします。キープアライブコマンドは、UDP I/Oポートに対して送信し、ここでは、前回設定したタイムアウト期間800ミリ秒より短い間隔で送信します。

パソコン→OS32C	
データグラム構造	UDP I/Oキープアライブ接続
サイズ	128バイト
更新パラメータ	EIP_UINT16 item_count = 0x0002 EIP_UINT16 address_item_type = 0x8002 EIP_UINT16 address_item_length = 0x0008 EIP_UINT32 o_to_t_connection_id = 0x780C0002 (OS32Cにより選択) EIP_UINT32 o_to_t_sequence_id = 0x00000001 (発信元デバイスにより選択) EIP_UINT16 data_item_type = 0x00B1 EIP_UINT16 data_item_length = 0x006E (出力アセンブリサイズ + 6) EIP_UINT16 o_to_t_16bit_sequence_# = 0x0001 (発信元デバイスにより選択) EIP_UINT32 header_trigger = 0x00000003 EIP_UINT8 DataArray [104] = 出力アセンブリ113データ 注意： 1) 発信元アプリケーションによるo_to_t_sequence_#のインクリメントが必要。 2) この例では、選択した計測ビーム数は20のみ。
16進法	02 00 02 80 08 00 02 00 0C 78 01 00 00 00 B1 00 6E 00 01 00 03 00 00 00 01 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 FF FF 0E 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

OS32C →パソコン	
データグラム構造	UDP I/Oキープアライブ接続
サイズ	0バイト
UDP I/Oキープアライブ接続応答例	
16進法	なし。このコマンドに対する応答はありません。この接続のままI/Oデータのストリーミングが継続されます。

10.2.2.10 対象（OS32C）および発信元（パソコン）から送信されるI/O接続アセンブリデータ

UDP I/O出力アセンブリによるトリガを送り、I/O接続をキープアライブし、新しいデータがあれば更新するようにします。キープアライブコマンドは、UDP I/Oポートに対して送信し、ここでは、前回設定したタイムアウト期間800ミリ秒より短い間隔で送信します。

OS32C→パソコン	
データグラム構造	UDP I/O接続ストリーミング
サイズ	116バイト
更新パラメータ	EIP_UINT16 item_count = 0x0002 EIP_UINT16 address_item_type = 0x8002 EIP_UINT16 address_item_length = 0x0008 EIP_UINT32 t_to_o_connection_id = 0x00020004 (OS32Cにより選択) EIP_UINT32 t_to_o_sequence_id = 0x00000015 (OS32Cにより選択) EIP_UINT16 data_item_type = 0x00B1 EIP_UINT16 data_item_length = 0x0062 (入力アセンブリサイズ + 2) EIP_UINT16 t_to_o_16bit_sequence_# = 0x00A1 (OS32Cにより選択) EIP_UINT8 DataArray [data_item_length - 2] = 入力アセンブリデータ (計測レポートヘッダ56バイト + ビームデータ40バイト) 注意： 1) OS32Cによってt_to_o_sequence_idのインクリメント。 2) この例では、20 (0x0014) 計測ビームのみ選択。
16進法	02 00 02 80 08 00 04 00 02 00 15 00 00 00 B1 00 62 00 A1 00 76 53 04 00 64 96 00 00 18 BE 97 8A 19 A7 00 00 03 00 07 00 00 00 00 00 00 00 01 00 00 00 08 07 88 33 AE 31 00 00 00 00 00 00 00 00 00 00 01 00 02 00 00 00 14 00 52 08 42 08 52 08 40 08 52 08 40 08 53 08 58 08 52 08 40 08 58 08 58 08 58 08 5E 08 67 08 5D 08 67 08 5E 08 5E 08 6F 08

10.2.2.11 発信元（パソコン）とOS32C間のForward Open I/O接続を終了する

セクション9.4.10に記載のSendRRData コマンドとセクション9.4.19のForward Close 要求を用いて、OS32CとのI/O接続を終了します。このコマンドは、スキャナセッションの登録を解除し、TCPソケット接続を閉じる前に送信します。

パソコン→OS32C	
データグラム構造	セクション9.4.11 表20：SendRRDataコマンド要求を参照 セクション9.4.19 表28：Forward Close要求のカプセル化パケットを参照
サイズ	62バイト
更新パラメータ	EIP_UINT16 length = 0x0026 (コマンド固有データの長さ) EIP_UINT32 session_handle_id = 0x00000005 EIP_UINT32 interface_handle_id = 0x00000000 EIP_UINT16 timeout = 0x0028 (40ミリ秒) EIP_UINT16 data_item_length = 0x0016 (22バイト) EIP_UINT8 service_request = 0x4E (Forward Close) EIP_UINT8 timeout_ticks = 0x28 (40ミリ秒) EIP_UINT16 connection_serial_number = 0x6789 (ユーザー選択可) EIP_UINT16 originator_vendor_id = 0x0195 (405, ORT) EIP_UINT32 originator_serial_number = 0x00004321 (ユーザー選択可)
16進法	<u>6F</u> 00 <u>26</u> <u>00</u> <u>05</u> <u>00</u> <u>00</u> <u>00</u> 00 00 00 00 00 00 00 00 00 00 00 00 00 00 <u>00</u> <u>00</u> <u>00</u> <u>00</u> <u>28</u> <u>00</u> 02 00 00 00 00 00 00 B2 00 <u>16</u> <u>00</u> <u>4E</u> 02 20 06 24 01 06 <u>28</u> <u>65</u> <u>87</u> <u>95</u> <u>01</u> <u>21</u> <u>43</u> 00 00 02 00 20 02 24 01

OS32C →パソコン	
データグラム構造	セクション9.4.20 表29：Forward Close要求に対するSendRRDataコマンド応答を参照
サイズ	54バイト
Forward Closeに対するSendRRData応答例	
返される 更新パラメータ	EIP_UINT16 length = 0x001E (コマンド固有データの長さ) EIP_UINT32 session_handle_id = 0x00000005 EIP_UINT16 data_item_length = 0x000E (14バイト) EIP_UINT8 service_response = 0xCE (Forward Close 0x80) EIP_UINT16 connection_serial_number = 0x6789 EIP_UINT16 originator_vendor_id = 0x0195 (405, ORT) EIP_UINT32 originator_serial_number = 0x00004321 EIP_UINT8 application_reply_size = 0x00 (16ビットワードの数) EIP_UINT8 reserved = 0x00
16進法	<u>6F</u> 00 <u>1E</u> <u>00</u> <u>05</u> <u>00</u> <u>00</u> <u>00</u> 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 02 00 00 00 00 00 00 00 B2 00 <u>0E</u> <u>00</u> <u>CE</u> 00 00 00 <u>65</u> <u>87</u> <u>95</u> <u>01</u> <u>21</u> <u>43</u> <u>00</u> <u>00</u> 00 00

11. マニュアル改訂履歴

マニュアル改訂記号は、表紙と裏表紙の下に記載されているカタログ番号の末尾に付記されます。

Cat. No. SCHG-740E

↑
改訂記号

改訂記号	改訂年月	改訂内容
A	2013年8月	初版
B	2013年11月	- CJ2、NJ、およびその他のPLCとのインターフェースをサポートするために、アセンブリデータサイズを縮小した入力アセンブリ104～111と出力アセンブリ114～120を追加。 - 圧縮された検出距離データおよび反射率データと別の形式にコード化された防護エリアビットと警告エリアビットに対応するために、検出距離レポートフォーマットを追加。 - ユーザーがアセンブリデータの最初と最後に事前定義データを挿入できるように、入力アセンブリ104～111用のユーザータグオプションを追加。 - 軽微の修正と更新
C	2014年4月	- 入力アセンブリ100、101、および出力アセンブリ113、114、115に有効警告エリアを追加。 - 有効警告エリアセットの選択を追加。 - 軽微の修正と更新
D	2020年2月	- 「ご承諾事項」の更新。 - 「1. はじめに」にEtherNet/IPの概要を追加。
E	2020年5月	軽微修正

●本誌に記載のない条件や環境での使用、および原子力制御・鉄道・航空・車両・燃焼装置・医療機器・娯楽機械・安全機器、その他人命や財産に大きな影響が予測されるなど、特に安全性が要求される用途に使用される際には、当社の意図した特別な商品用途の場合や特別の合意がある場合を除き、当社は当社商品に対して一切保証をいたしません。
●本製品の内、外国為替及び外国貿易法に定める輸出許可、承認対象貨物(又は技術)に該当するものを輸出(又は非居住者に提供)する場合は同法に基づく輸出許可、承認(又は役務取引許可)が必要です。

オムロン株式会社 インダストリアルオートメーションビジネスカンパニー

●製品に関するお問い合わせ先

お客様相談室



0120-919-066

携帯電話・PHS・IP電話などではご利用いただけませんので、下記の電話番号へおかけください。

電話 055-982-5015 (通話料がかかります)

■営業時間：8:00～21:00

■営業日：365日

●FAXやWebページでもお問い合わせいただけます。

FAX 055-982-5051 / www.fa.omron.co.jp

●その他のお問い合わせ

納期・価格・サンプル・仕様書は貴社のお取引先、または貴社担当オムロン販売員にご相談ください。

オムロン制御機器販売店やオムロン販売拠点は、Webページでご案内しています。

オムロン制御機器の最新情報をご覧ください。

www.fa.omron.co.jp

緊急時のご購入にもご利用ください。

オムロン商品のご用命は

Man. No.

SCHG-740E

2020年5月現在

© OMRON Corporation 2013-2020 All Rights Reserved.

お断りなく仕様などを変更することがありますのでご了承ください