

# Anybus Communicator (ABC) Explicit メッセージによる Ethernet/IP と バーコードリーダーの接続

Version: A01



エイチエムエス・インダストリアルネットワークス株式会社  
〒222-0033

神奈川県横浜市港北区新横浜 3-19-5  
新横浜第2センタービル 6F

TEL : 045-478-5340

FAX : 045-476-0315

URL

[www.anybus.jp](http://www.anybus.jp)

EMAIL

セールス:[jp-sales@hms-networks.com](mailto:jp-sales@hms-networks.com)  
サポート:[jp-support@hms-networks.com](mailto:jp-support@hms-networks.com)

EVOLUTION OF THE DOCUMENT ..... 3

1. 前提 ..... 4

2. ABC コンフィグレーション例 ..... 4

2.1. 構成図 ..... 4

2.2. メッセージ（EXPLICIT）データーとしてのデーター転送 ..... 5

2.2.1. 例 1（全メモリー領域をメッセージデーター領域として使用） ..... 5

2.2.1. 例 2（例 1 に対してアトリビュートを追加） ..... 11

2.2.1. 例 3（I/O データー領域とメッセージデーター領域の両方を使用） ..... 12

**EVOLUTION OF THE DOCUMENT**

---

Issue	Date	Author	Motive and nature of the modifications
A01	2011/09/06	KAH	First release.

---

---

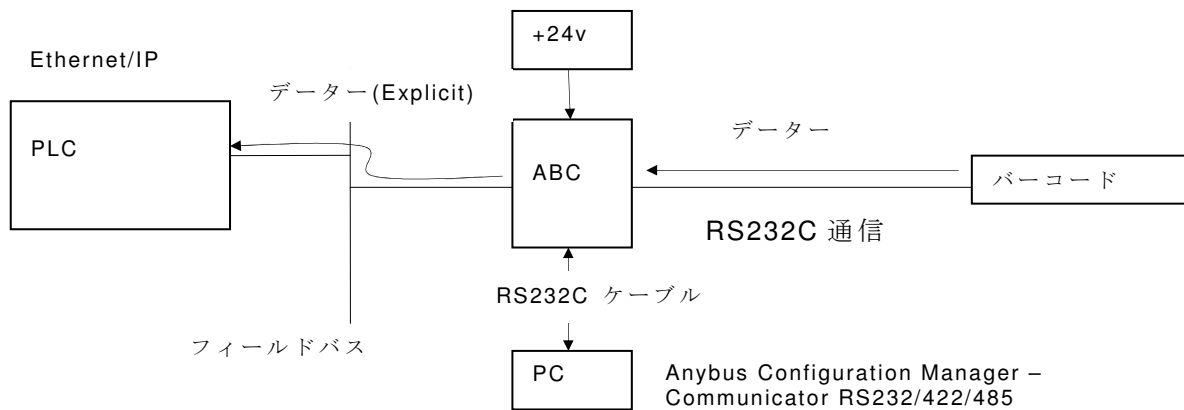
This document contains: 16 pages.

## 1. 前提

本ドキュメントは、“Anybus Communicator(ABC)サイクリックデータによる Ethernet/IP とバーコードリーダーの接続”ドキュメントをベースに Explicit メッセージ転送に必要な部分のみを記述しています。 よって、本ドキュメントを使用する前に “Anybus Communicator(ABC)サイクリックデータによる Ethernet/IP とバーコードリーダーの接続” で基本的な ABC の設定を理解した上で御使用をお願い致します。

## 2. ABC コンフィグレーション例

### 2.1. 構成図

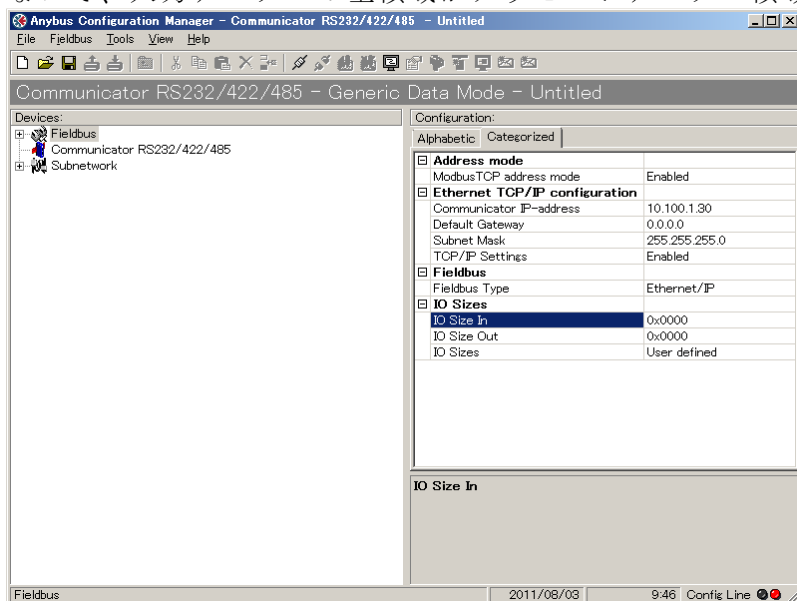


## 2.2. メッセージ（Explicit）データとしてのデータ転送

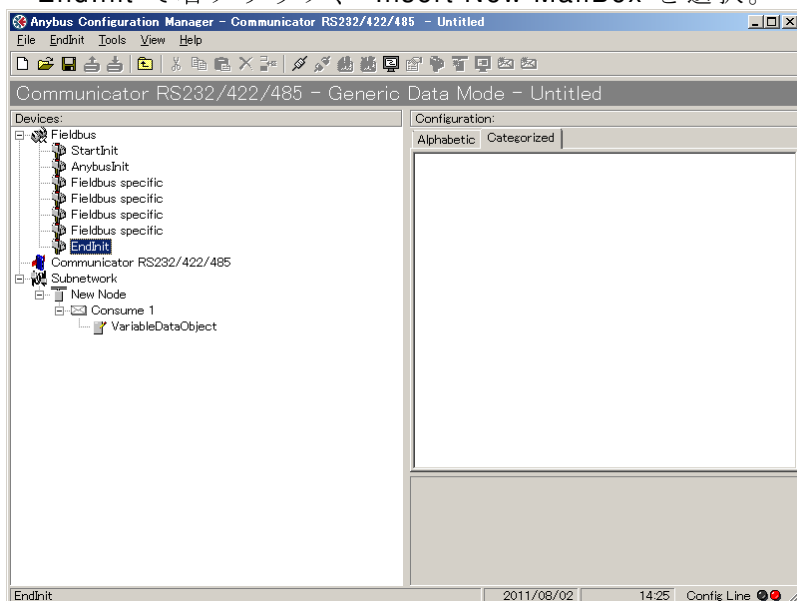
先に”Anybus Communicator(ABC)サイクリックデータによる Ethernet/IP とバーコードリーダーの接続”ドキュメント、及び”User Manual Anybus® Communicator™ for EtherNet/IP”ドキュメント、特に”Parameter Data Input Mapping Object, Class B0h”、“Class B1h”、及び”Appendix A”項を理解後、本ドキュメントを御使用下さい。

### 2.2.1. 例 1（全メモリー領域をメッセージデータ領域として使用）

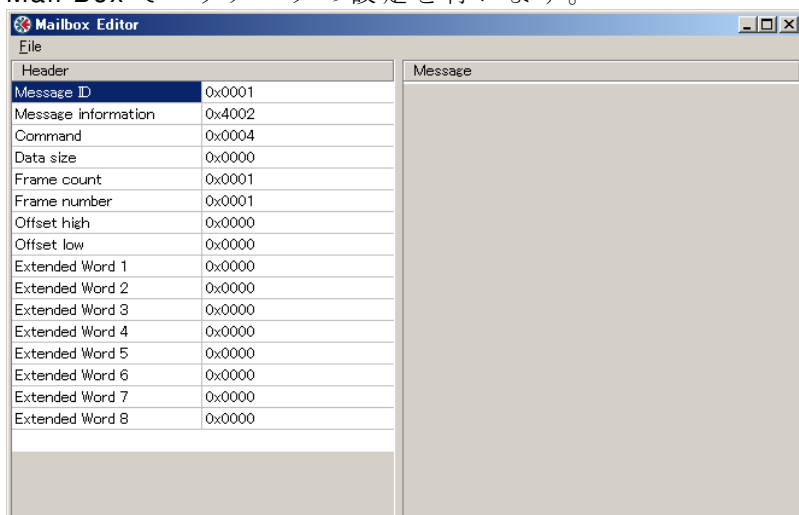
“Anybus Communicator(ABC)サイクリックデータによる Ethernet/IP とバーコードリーダーの接続の”フィールドバス側の設定”の項に従ってフィールドバスの設定を行います。その後、IO Size を”User Defined”にし、”IO Size in”を 0 にします。これは IO データ入力としては使用しないで、入力データの全領域がメッセージデータ領域となることを意味します。



“EndInit”で右クリック、“Insert New MailBox”を選択。



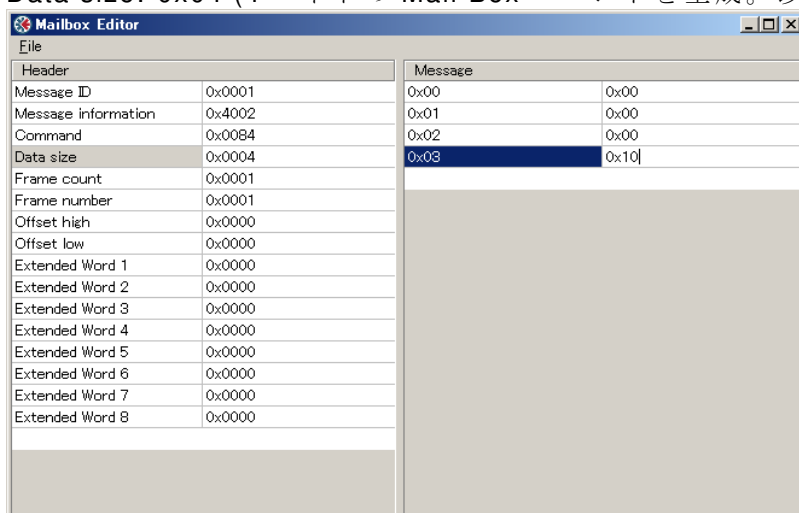
Mail Box でパラメータの設定を行います。



設定内容は、

Command: 0x84 (仕様で規定)。

Data size: 0x04 (4 バイトの Mail Box コマンドを生成。以下 Message 部 0x00~0x03 まで)。

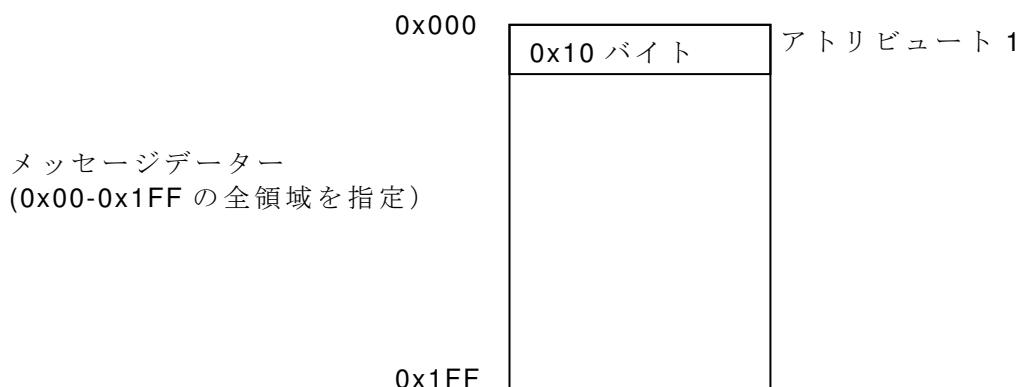


メッセージデーターメモリー領域の先頭からのオフセット値を指定

メッセージデーターとして使用するメモリーのデーターサイズ

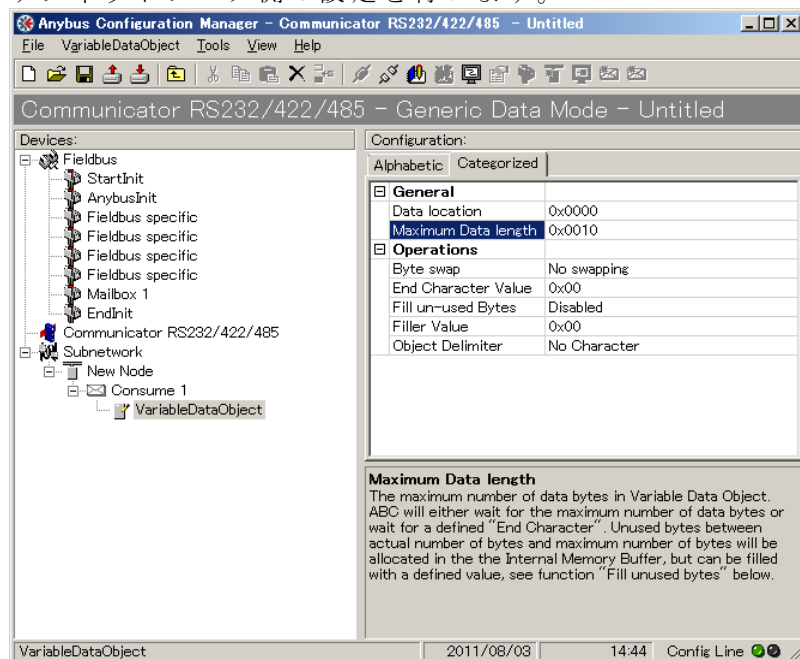
上記 Mail Box コマンド (Message 部) 0x00-0x01 により、ABC のメモリー領域において、メッセージデーターとしての先頭アドレスからのオフセット値を指定します。残りの 0x02-0x03 ではメッセージデーターとしてのサイズを指定します。

上記例は、ABC で使用するメモリー領域すべて (0x00-0x1FF) をメッセージデーターとして定義し、その先頭 10 バイトを使用 (以下参照)。



メニューバー”File”の”Apply Changes”を実行。

サブネットワーク側の設定を行います。



先の指定で、メッセージデータとして使用するメモリのデータサイズを **0x10** としたので、**0x10** 以上のデータサイズの指定をする必要がある。今回は **0x10** を指定。

上記設定完了後、ダウンロードを行います。

EIP Scan で次の設定を行います。

#### Target

Network Path: 10.200.1.30 <= ABC の IP アドレス  
Adapter : 10.200.1.20 <= PLC 側 (PC 側) の IP アドレス

#### Class 1 connection

##### Transport Type

Originator -> Target: Point to Point

Target -> Originator: Multicast

##### Data Size

Originator -> Target: 0

Target -> Originator: 0 <=I/O データー通信は行わないので 0 とします

##### Rate

###### Packet Rate in milliseconds

Originator -> Target: 100

Target -> Originator: 100

###### Production Inhibit Timeout in milliseconds

Originator -> Target: 0

Target -> Originator: 0

##### Trigger

Transport Trigger: Cyclic

Timeout Multiplier: 16

##### Destination

Configuration Connection Instance: 1

Originator -> Target - Specify Connection Point or Tag

Connection point: 150 <=ABC のデフォルト値

Target -> Originator - Specify Connection Point or Tag

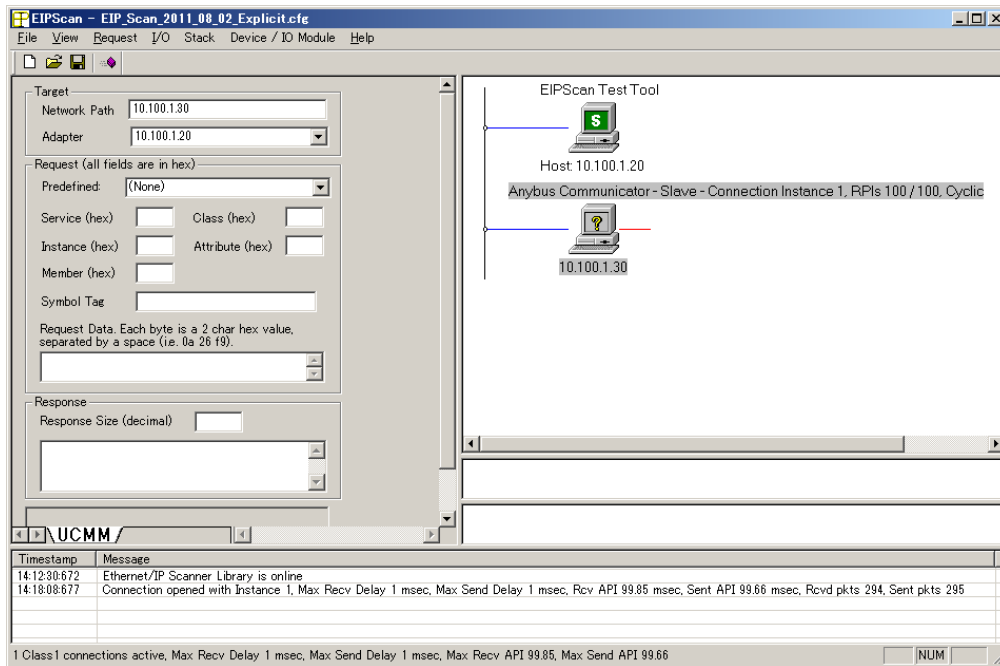
Connection point: 100 <=ABC のデフォルト値

##### Priority

Originator -> Target: Scheduled

Target -> Originator: Scheduled

EIP Scan を実行します。



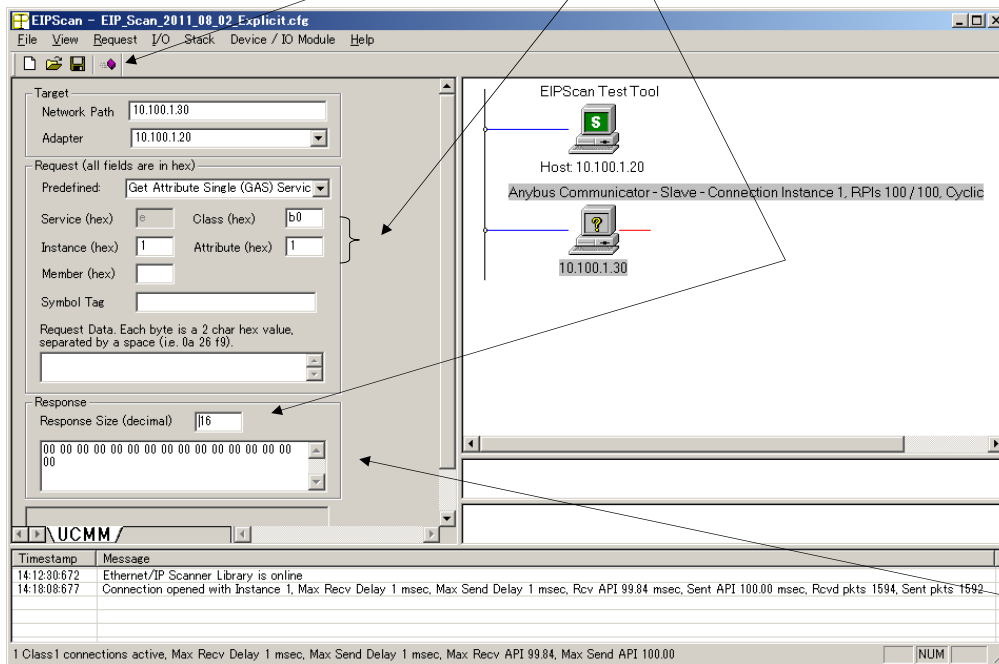


ここで、”Get Attribute Single (GAS) Service”を選択。以下の CIP のファンクションを設定します。

Class :           b0 (オブジェクト番号)  
Instance :        1 (インスタンス)  
Attribute :       1 (アトリビュート)

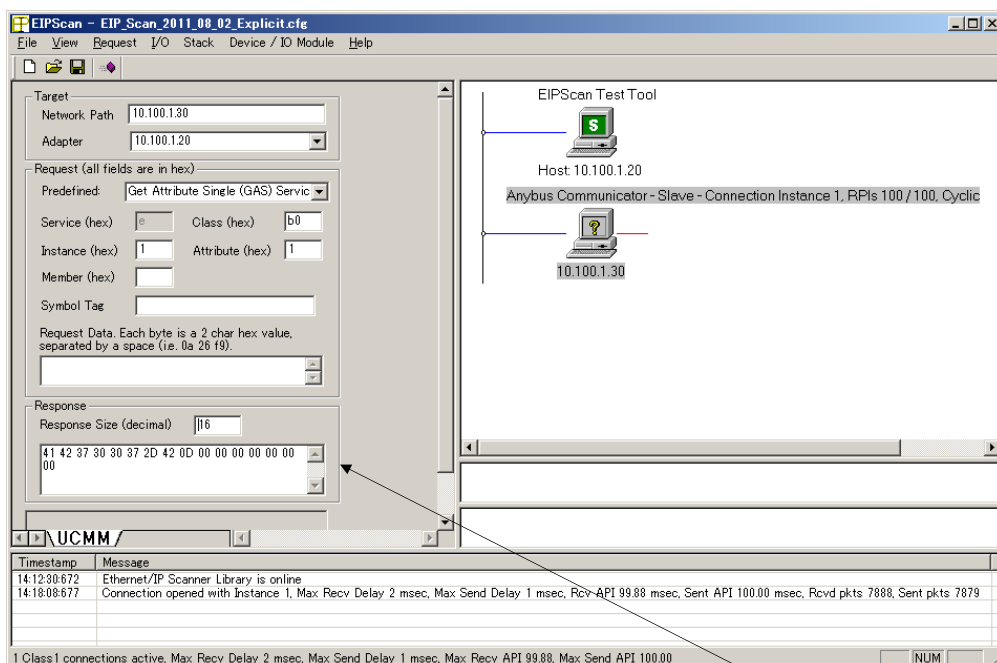
Response Size: 16 (取得データサイズ)

その後、実行します。



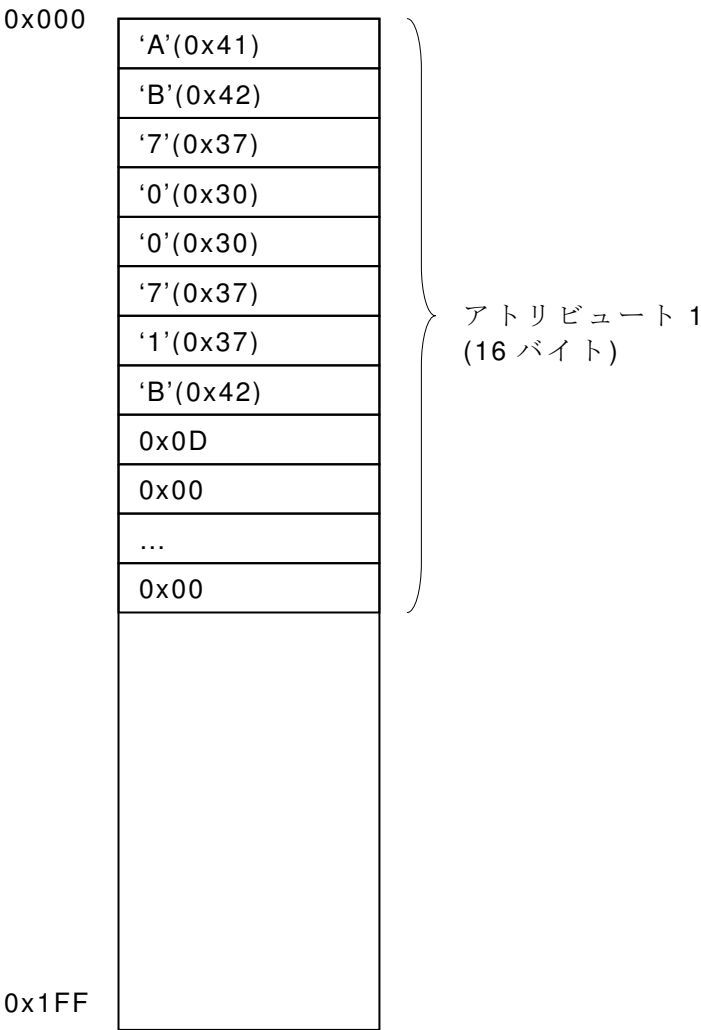
まだ、バーコードより読み込みのデータがないので、すべて 0 になっています。

バーコードより読み込み後、再度データの取得を行います。



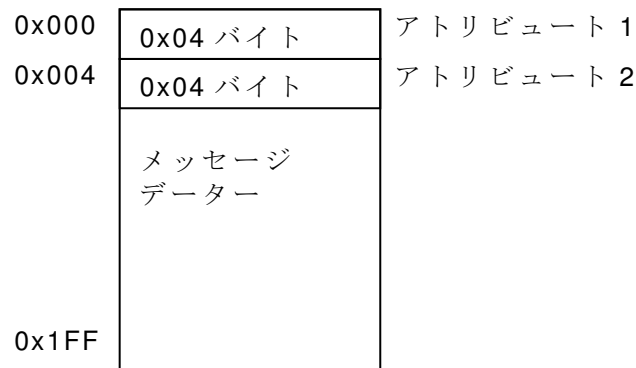
データが読み込まれていることがわかります。

以下はバーコードよりデータを読み込んだ場合のマップを示します（例）。



### 2.2.1. 例 2（例 1 に対してアトリビュートを追加）

上記例 1 に対して、アトリビュート 2 を追加してみる。 オフセット 0、データーサイズ 0x04 をアトリビュート 1、オフセット 0x04、データーサイズ 0x04 をアトリビュート 2 として定義します。

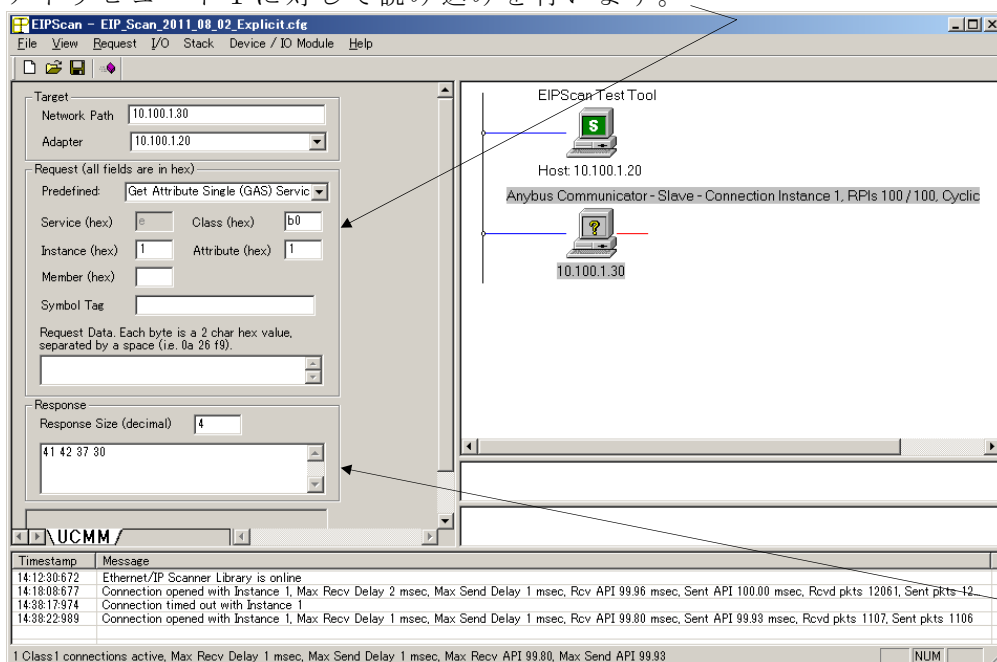


以下上記例 1 からの変更部分のみ記載します。

Header		Message	
Message ID	0x0001	0x00	0x00
Message information	0x4002	0x01	0x00
Command	0x0084	0x02	0x00
Data size	0x0008	0x03	0x04
Frame count	0x0001	0x04	0x00
Frame number	0x0001	0x05	0x04
Offset high	0x0000	0x06	0x00
Offset low	0x0000	0x07	0x04
Extended Word 1	0x0000		
Extended Word 2	0x0000		
Extended Word 3	0x0000		
Extended Word 4	0x0000		
Extended Word 5	0x0000		
Extended Word 6	0x0000		
Extended Word 7	0x0000		
Extended Word 8	0x0000		

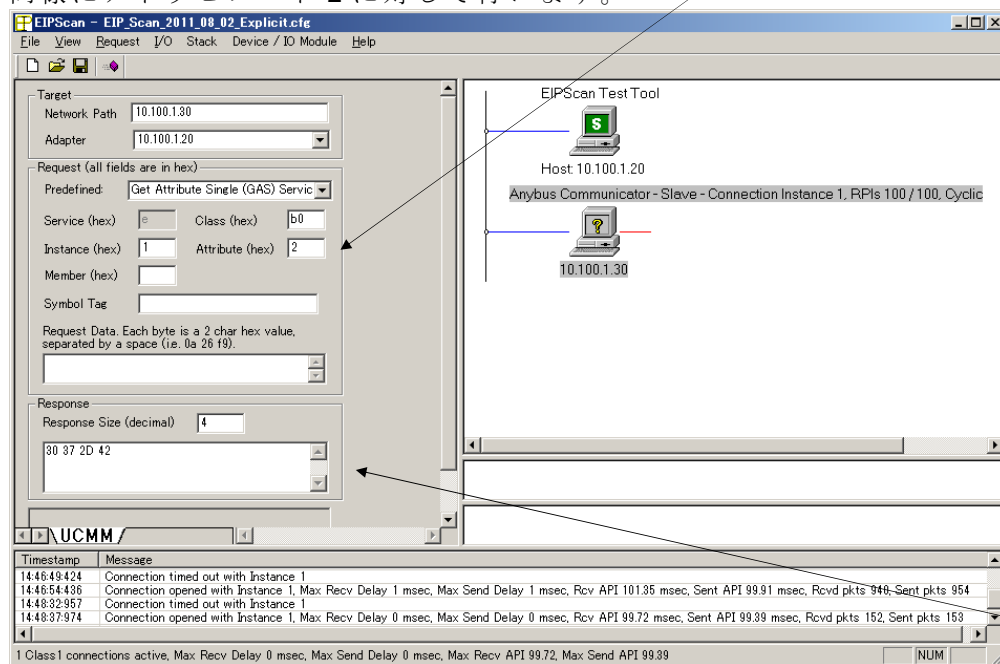
0x00 番地（オフセット）アトリビュート 1  
データー領域（0x04）  
0x04 番地（オフセット）アトリビュート 2  
データー領域（0x04）

アトリビュート 1 に対して読み込みを行います。



メモリー上の 4 バイト “AB70”（0x41,0x42,0x37,0x30）が読み込まれます。

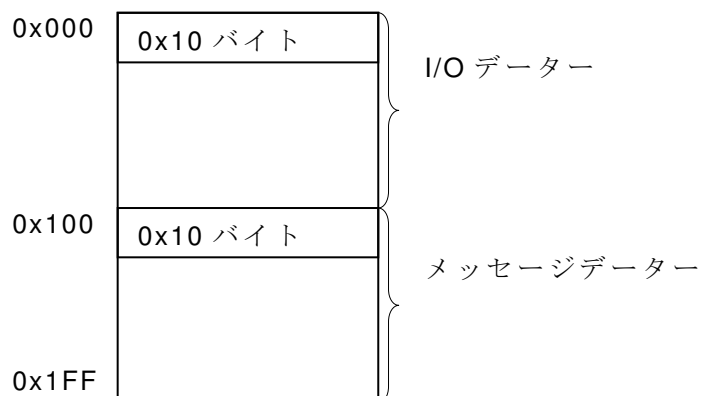
同様にアトリビュート 2 に対して行います。



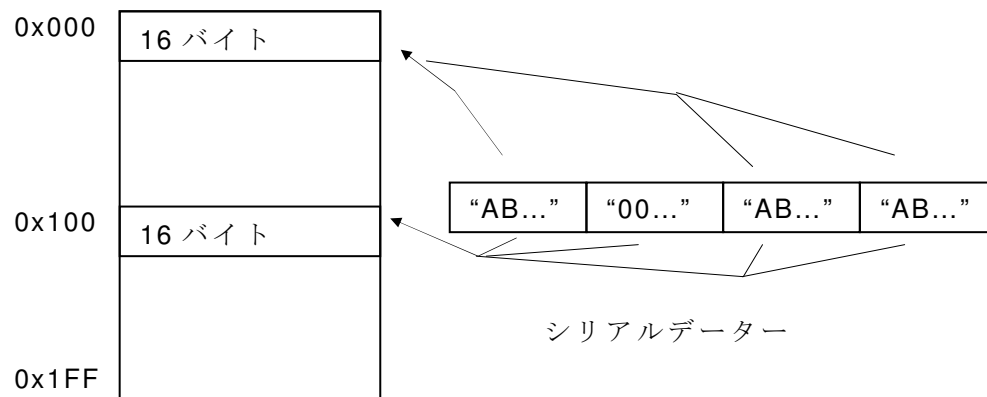
メモリー上の 4 バイト “071B” (0x30,0x37,0x2D,0x42) が読み込まれます。

### 2.2.1. 例 3 (I/O データー領域とメッセージデーター領域の両方を使用)

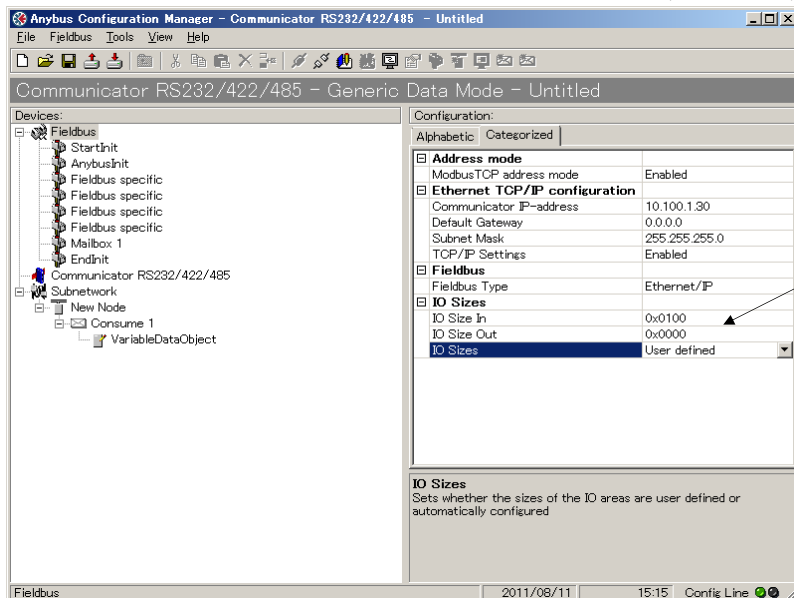
I/O データーとして 0x000-0x0FF、メッセージデーターとして 0x100-0x1FF を使用。



又、シリアル側からのデータ入力として、先頭キャラクターが'A' (0x41) できた場合のみ I/O データ領域へ入力されるようにします。 又、メッセージデータ領域にはすべてのデータを入力するように設定します。 データとして“AB00、”を読み込んだ場合、I/O データとメッセージデータ領域の両方に入力されます。



I/O サイズとして“IO Size In”を 0x100 バイトと指定します。



上記の設定により、0x00 番地から 0xFF 番地までが I/O データ領域となり、残りの 0x100 番地からがメッセージデータ領域となります。

メッセージサイズで 16 バイトを指定。

Header	Message
Message ID	0x0001
Message information	0x4002
Command	0x0084
Data size	0x0004
Frame count	0x0001
Frame number	0x0001
Offset high	0x0000
Offset low	0x0000
Extended Word 1	0x0000
Extended Word 2	0x0000
Extended Word 3	0x0000
Extended Word 4	0x0000
Extended Word 5	0x0000
Extended Word 6	0x0000
Extended Word 7	0x0000
Extended Word 8	0x0000

以後はシリアル側の設定を行います。先頭番地が'A'(0x41)の時に処理されるように **Consume1** を設定します。

Configuration: Alphabetic Categorized

General

Value 0x41

Value  
Definition of a Constant Byte Value. When transmitting the Byte this Value will be sent. When receiving a Byte ABC will perform a check against this Value. If the received Byte does not match it will be discarded.

**Consume1** は先頭キャラクターが'A'(0x41)の場合に処理されるように、Byte Constant を指定します。

Configuration: Alphabetic Categorized

General

Data location 0x0000

Maximum Data length 0x0010

Operations

Byte swap No swapping

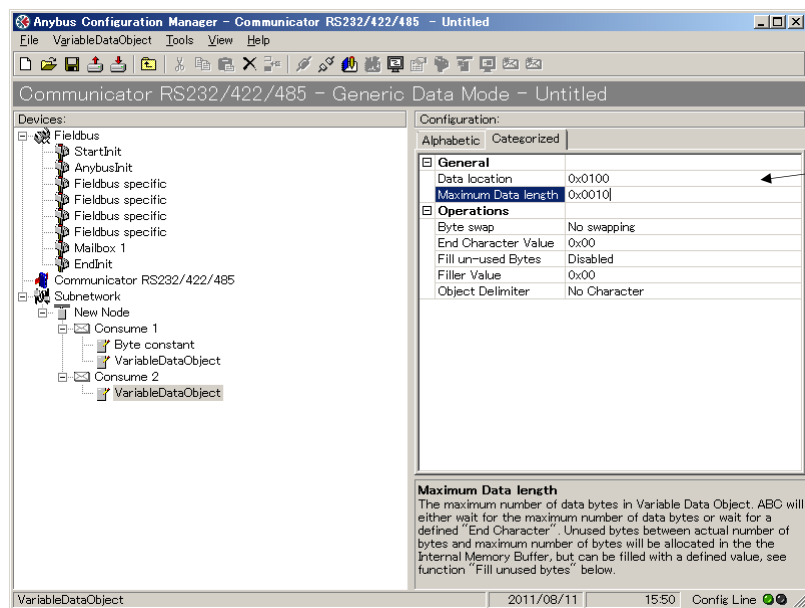
End Character Value 0x00

Fill un-used Bytes Disabled

Filler Value 0x00

Object Delimiter No Character

先頭キャラクターが'A'(0x41)の場合 0x0 番地よりデータ入力行われます。



Consume2 はすべてのデータが 0x100 番地より入力されます。

上記設定をダウンロード後、EIP Scan を実行します（EIP Scan の設定変更要）。

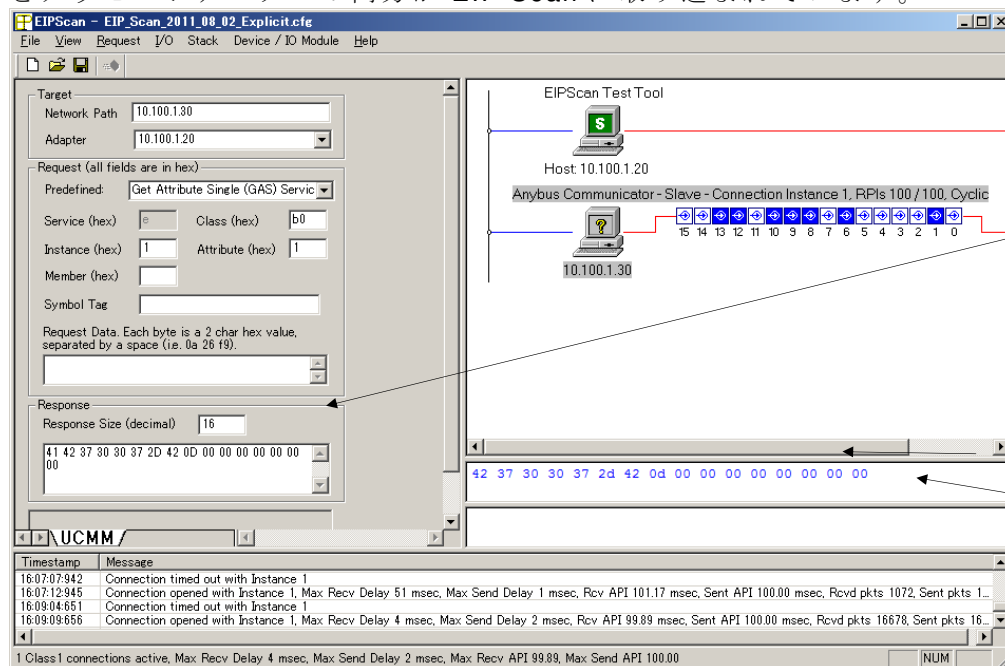
## Class 1 connection

```

...
Data Size
  Originator -> Target: 0
  Target -> Originator: 16  <=I/O データサイズ
Rate
...

```

バーコードリーダーより“AB7007-B”を読み込みます。以下に表示されるように、I/O データとメッセージデータの両方が EIP Scan に取り込まれています。



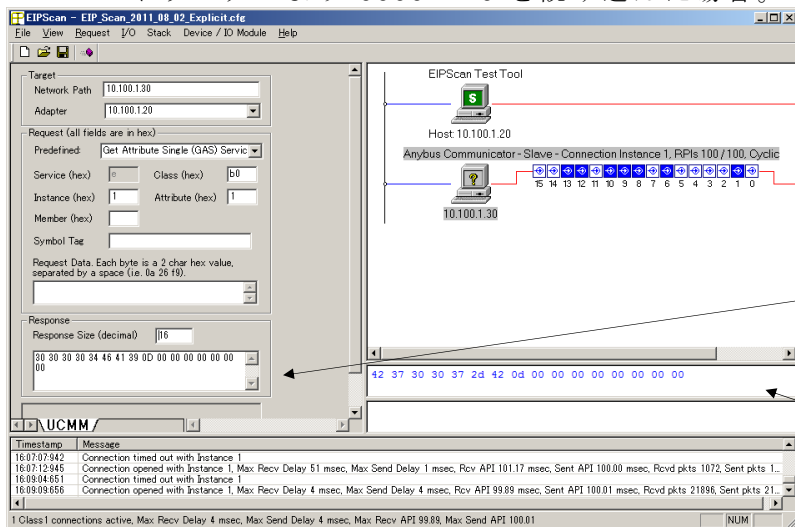
メッセージデータ  
("AB7007-B"+0x0D)  
が取り込まれていま  
す

I/O データ  
("B7007-B"+0x0D)  
が取り込まれていま  
す。

注意)

シリアル側から入力されるデータの先頭の'A'(0x41)は、“Byte Constant”として指定されている為に I/O データとしては認識されない。 よって、I/O データとしては次のデータ'B'よ(0x42)り取り込まれます。

バーコードリーダーより“00004FA9”を読み込んだ場合。



メッセージデーターとしては“00004FA9”が取り込まれています。

“00004FA9”を読み込んでも、先頭バイトが'A'(0x41)でない為に、データーが取り込まれません。

以上