Anybus Comunicator(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

EMAIL
セールス:jp-sales@hms-networks.com
サポート:jp-support@hms-networks.com

EVOLUTION OF THE DOCUMENT

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

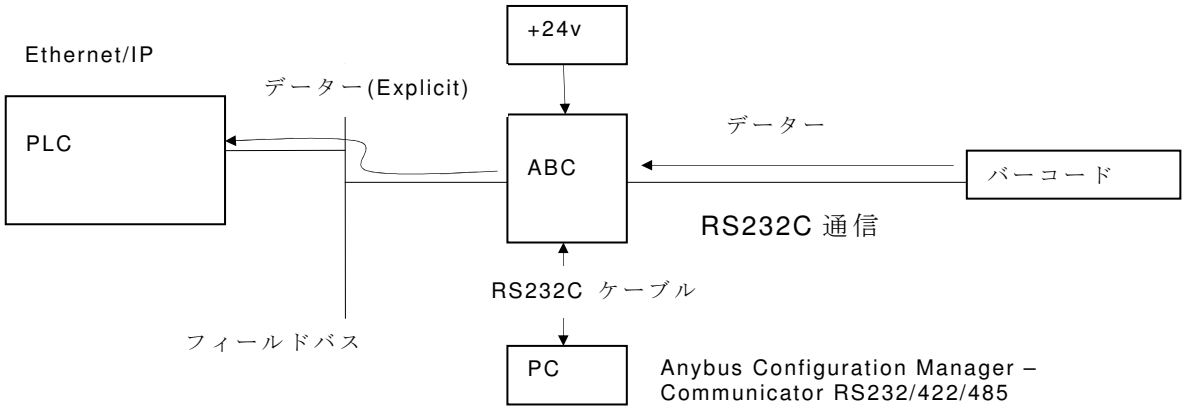
This document contains: 16 pages.

1. 前提

本ドキュメントは、“Anybus Comunicator(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 Comunicator(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

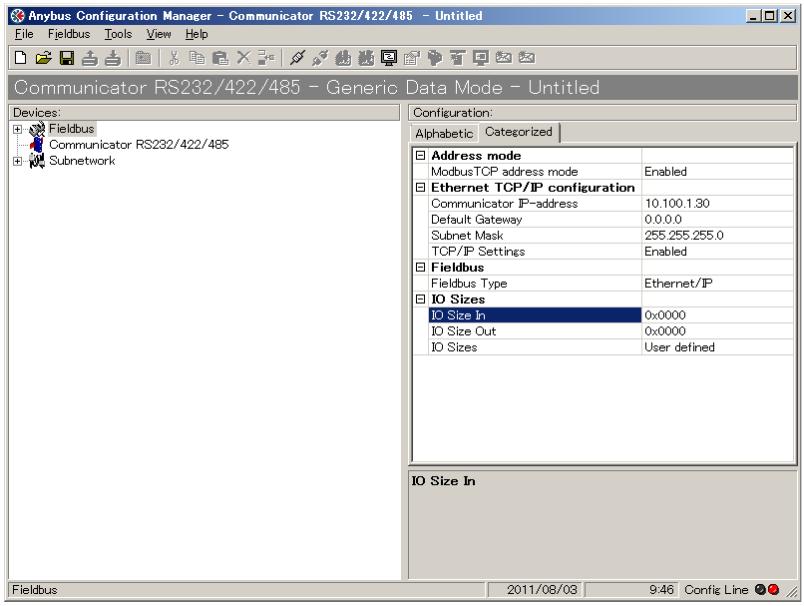


2.2. メッセージ（Explicit）データーとしてのデーター転送

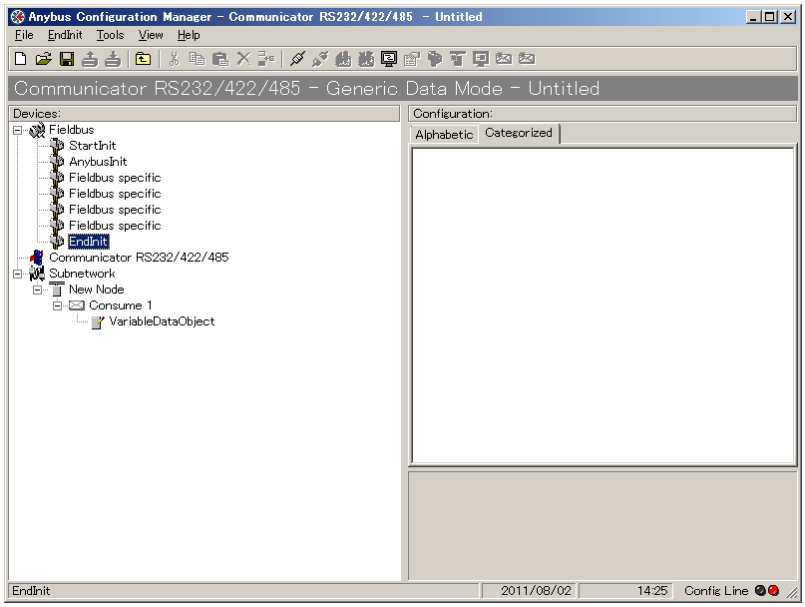
先に”Anybus Comunicator(ABC)サイクリックデータによる Ethernet/IP とバーコードリーダーの接続”**ドキュメント、及び**“User Manual Anybus® Communicator™ for EtherNet/IP” ドキュメント、特に” Parameter Data Input Mapping Object, Class B0h“、”ClassB1h” 、及び“AppendixA” **項を理解後、本ドキュメントを御使用下さい。**

2.2.1. 例 1 （全メモリー領域をメッセージデーター領域として使用）

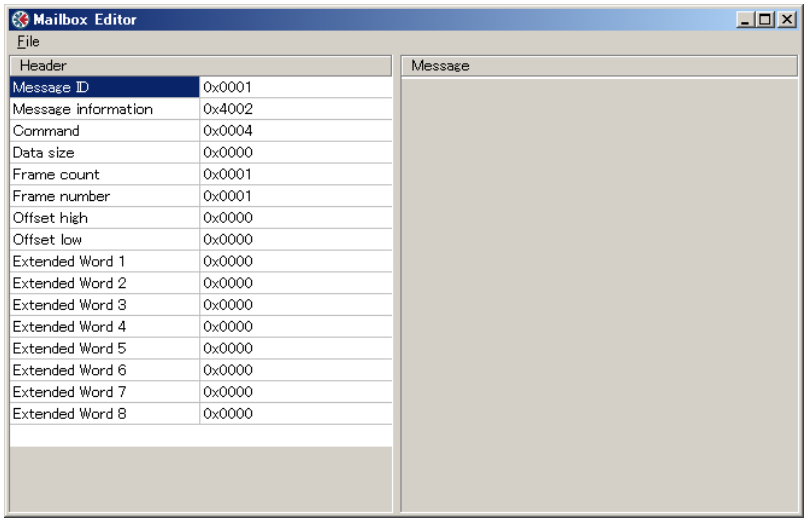
“Anybus Comunicator(ABC)サイクリックデータによる Ethernet/IP とバーコードリーダーの接続の” フィールドバス側の設定” の項に従ってフィールドバスの設定を行います。その後、IO Size を” User Defined” にし、” IO Size in” を 0 にします。これは IO データー入力としては使用しないで、入力データーの全領域がメッセージデーター領域となることを意味します。



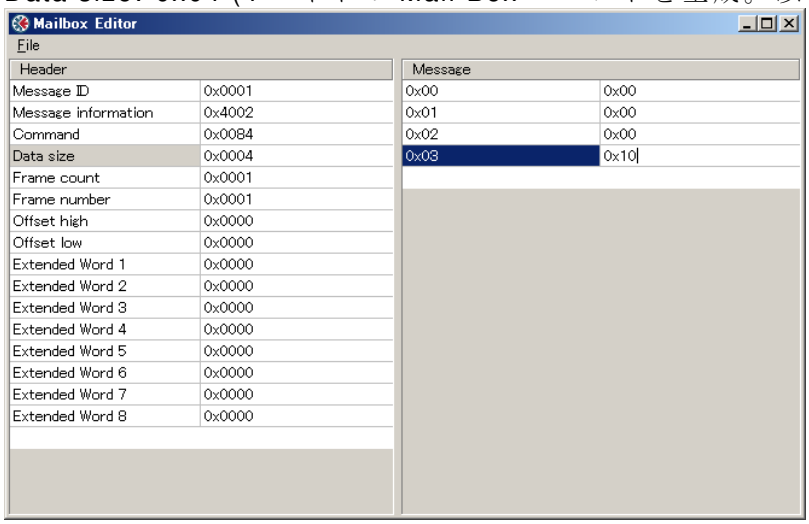
“EndInit”で右クリック、”Insert New MailBox”を選択。



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)。

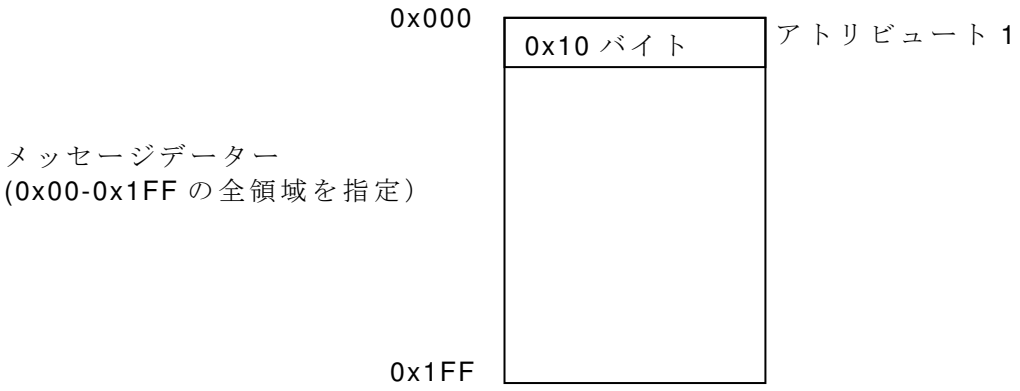


メッセージデーターメモリー領域の先頭からのオフセット値を指定

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

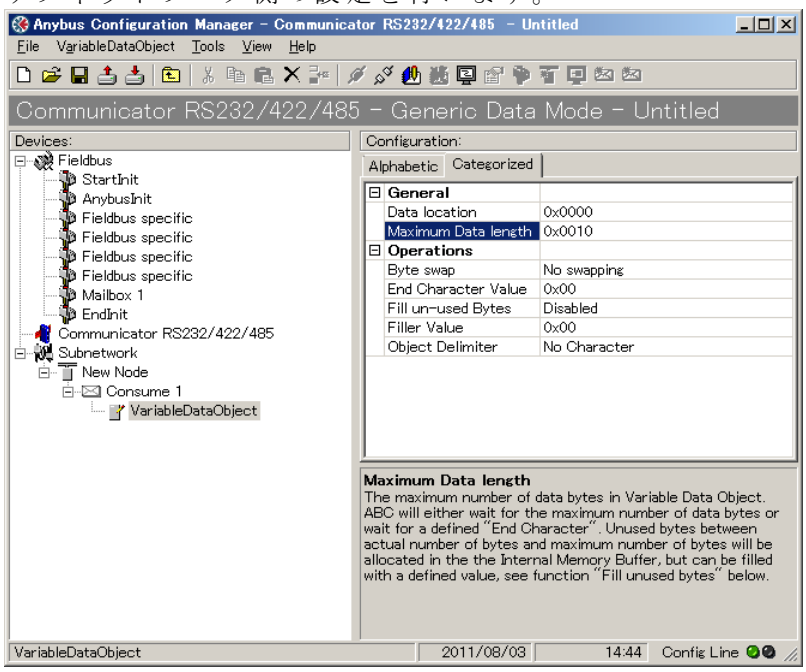
上記 Mail Box コマンド（Message 部）0x00-0x01 により、ABC のメモリー領域において、メッセージデーターとしての先頭アドレスからのオフセット値を指定します。　残りの 0x02-0x03 ではメッセージデーターとしてのサイズを指定します。

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、メッセージデーターとして使用するメモリーのデーターサイズを 0x10 としたので、0x10 以上のデーターサイズの指定をする必要がある。今回は 0x10 を指定。

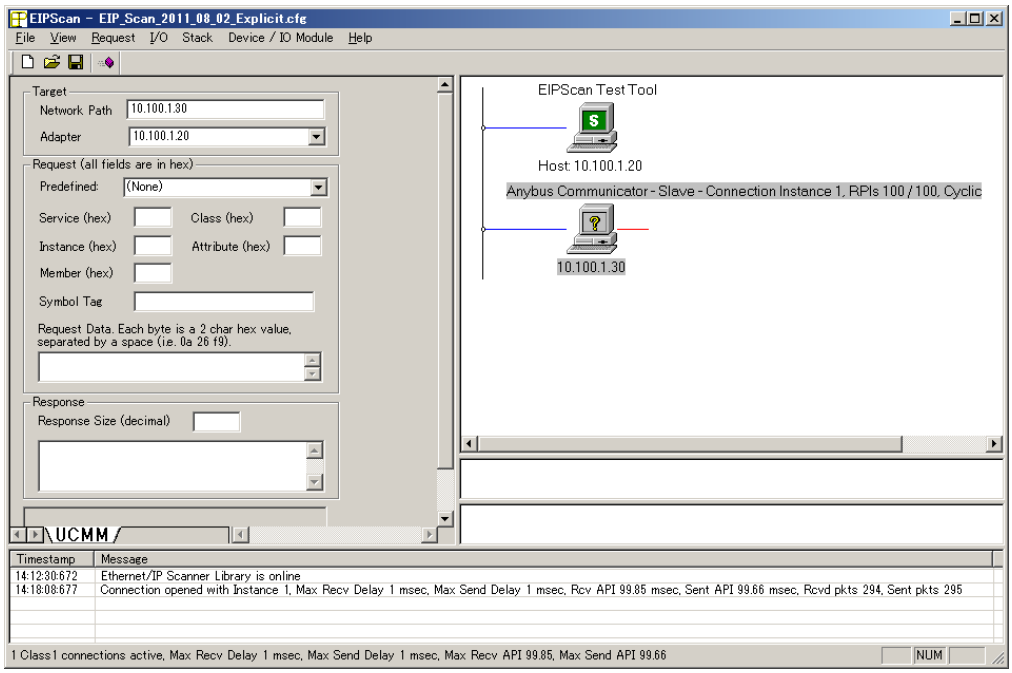
上記設定完了後、ダウンロードを行います。

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 のディフォルト値
  - Target -> Originator – Specify Connection Point or Tag
    - Connection point: 100 <=ABC のディフォルト値
- Priority
  - Originator -> Target: Scheduled
  - Target -> Originator: Scheduled

EIP Scan を実行します。

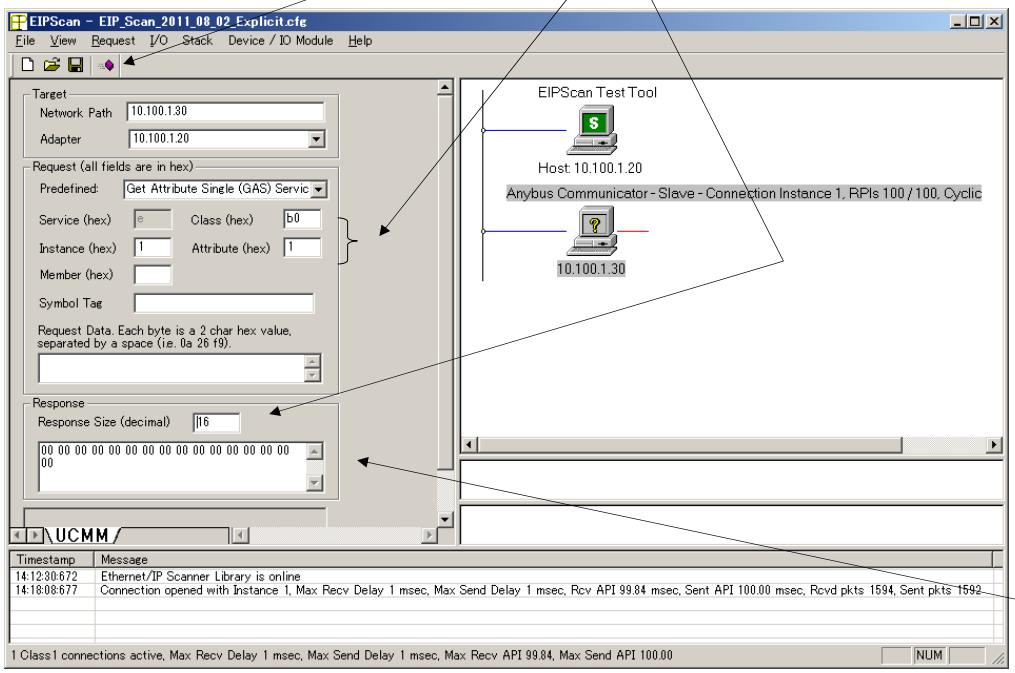


ここで、”Get Attribute Single (GAS) Service”を選択。以下の CIP のファンクションを設定します。

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）

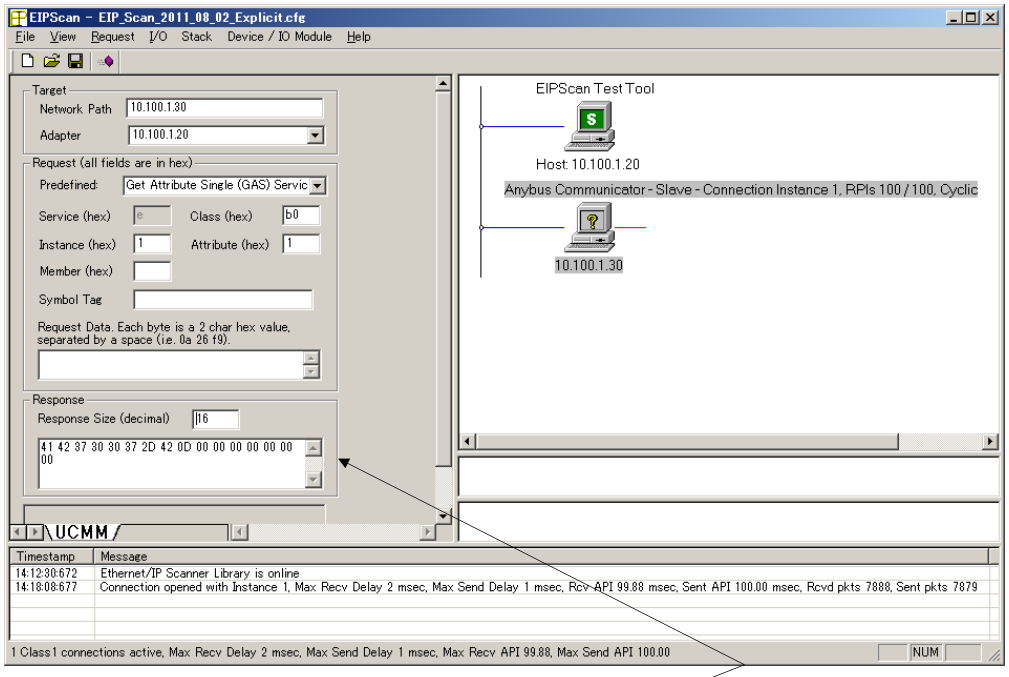
Response Size: 16(取得データーサイズ)

その後、実行します。



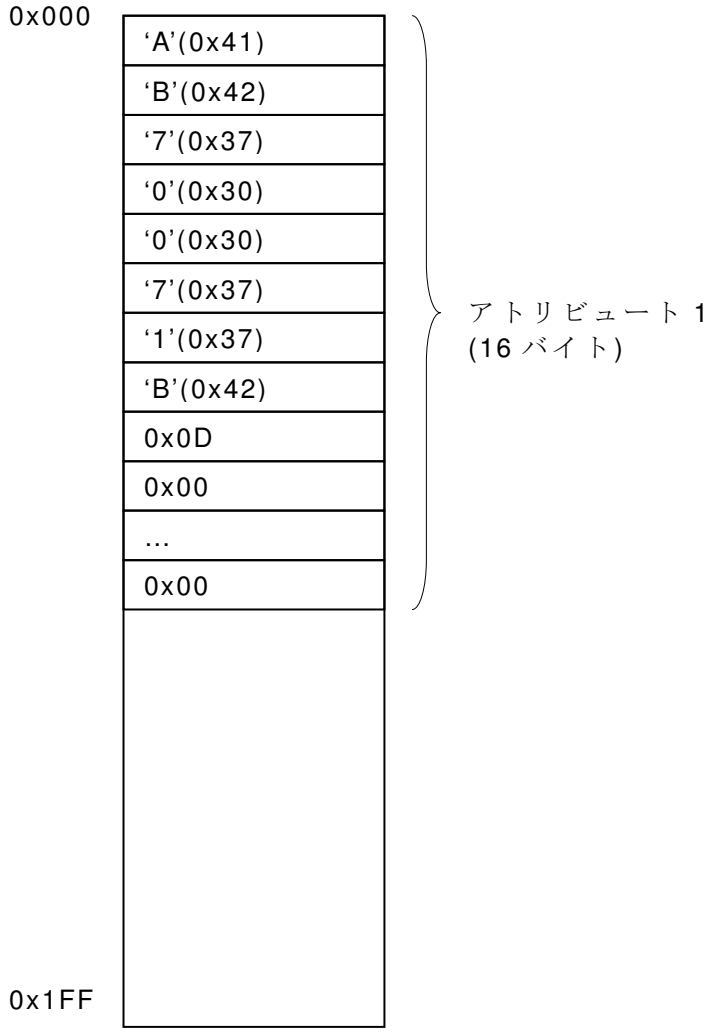
まだ、バーコードより読み込みのデーターがないので、すべて 0 になっています。

バーコードより読み込み後、再度データーの取得を行います。



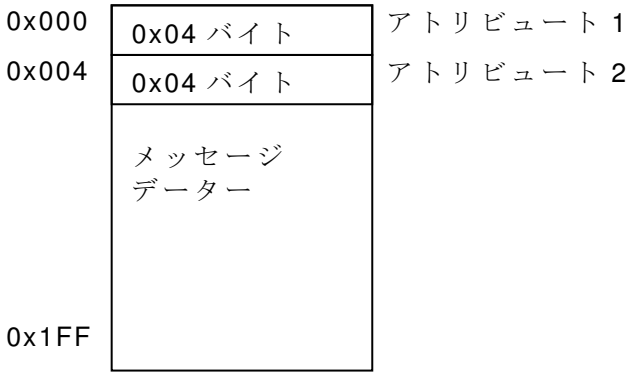
データーが読み込まれていることがわかります。

以下はバーコードよりデーターを読み込んだ場合のマップを示します（例）。

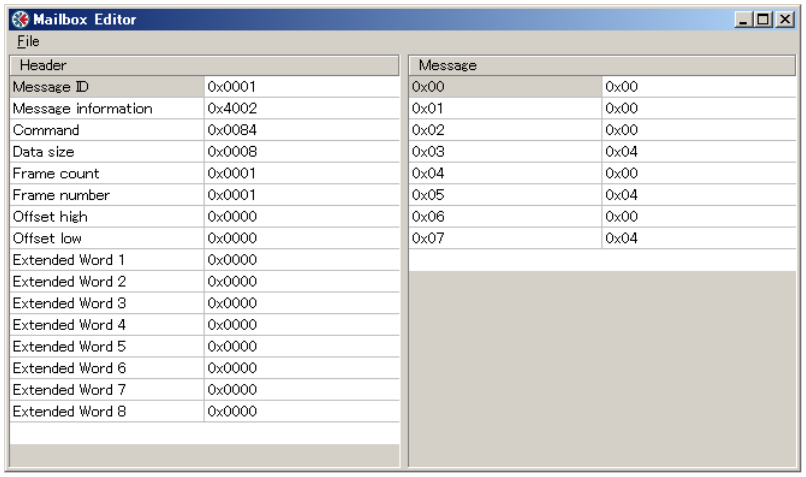


2.2.1. 例 2（例 1 に対してアトリビュートを追加）

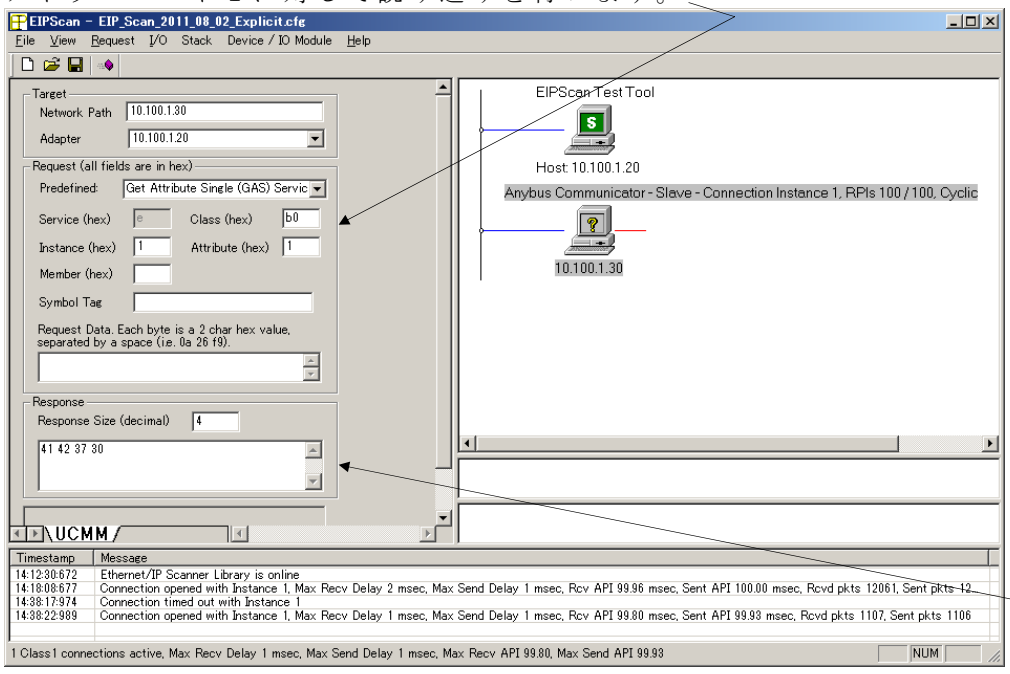
上記例 1 に対して、アトリビュート 2 を追加してみる。 オフセット 0、データーサイズ 0x04 をアトリビュート 1、オフセット 0x04、データーサイズ 0x04 をアトリビュート 2 として定義します。



以下上記例 1 からの変更部分のみ記載します。

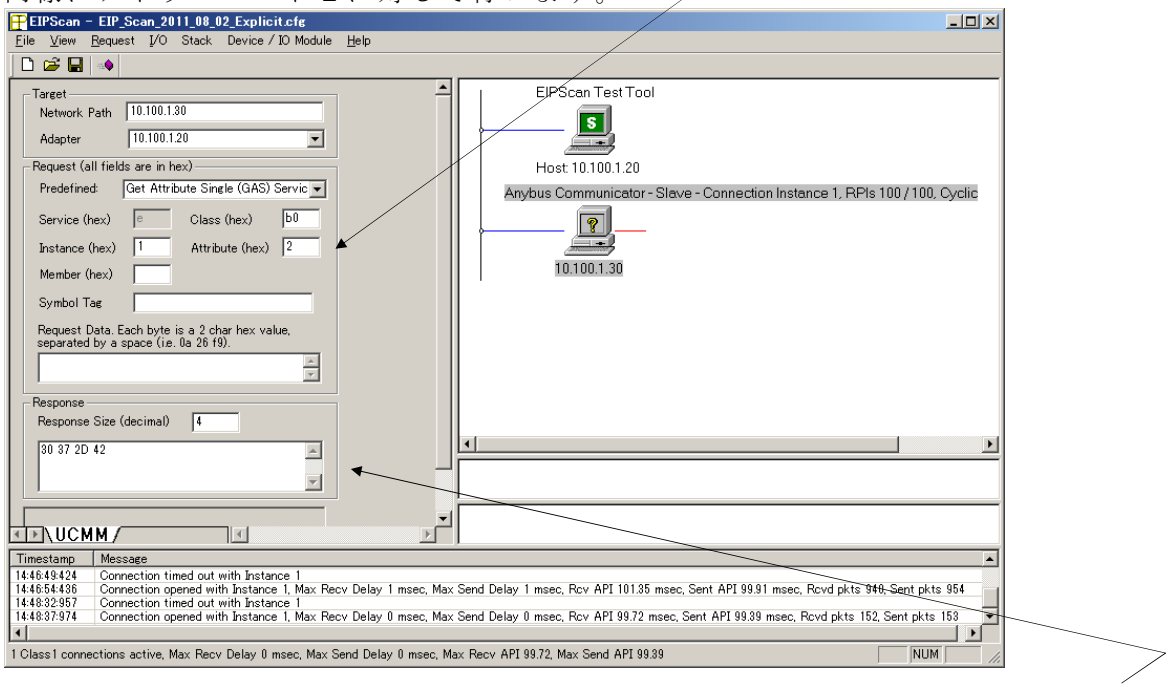


アトリビュート 1 に対して読み込みを行います。



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

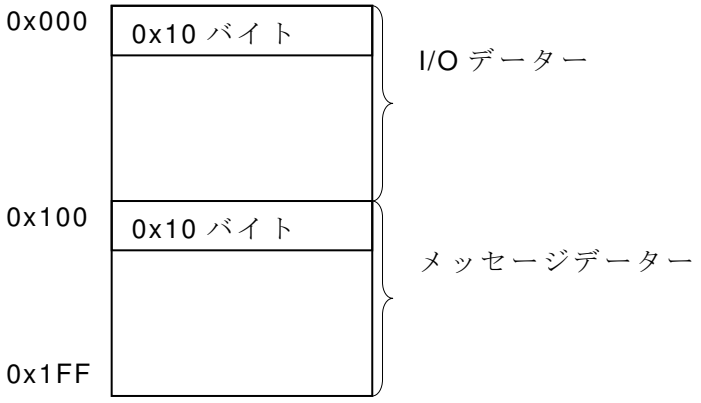
同様にアトリビュート 2 に対して行います。



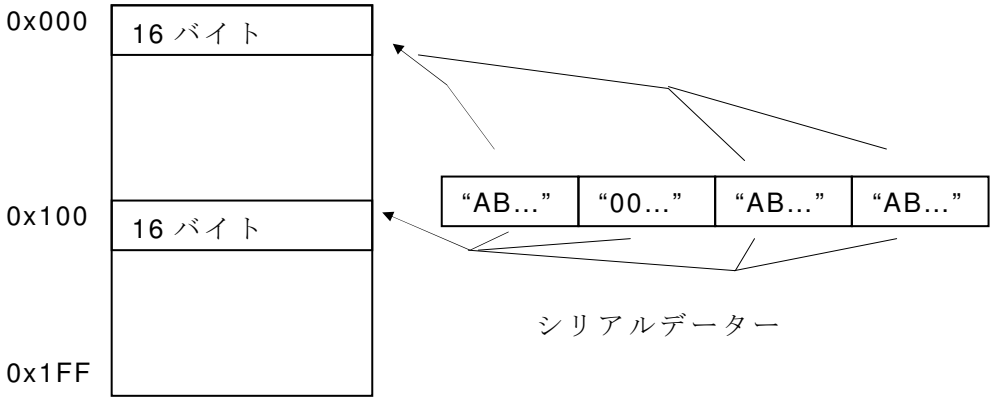
メモリー上の 4 バイト“071B”（0x30,0x37,0x2D,0x42）が読み込まれます。

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）

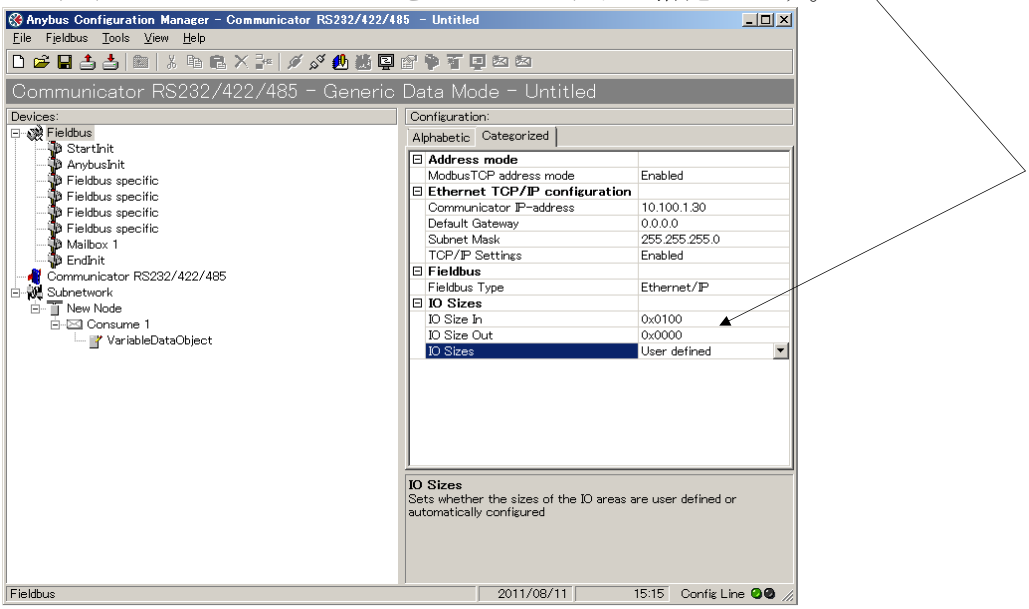
I/O データーとして 0x000-0x0FF、メッセージデーターとして 0x100-0x1FF を使用。



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

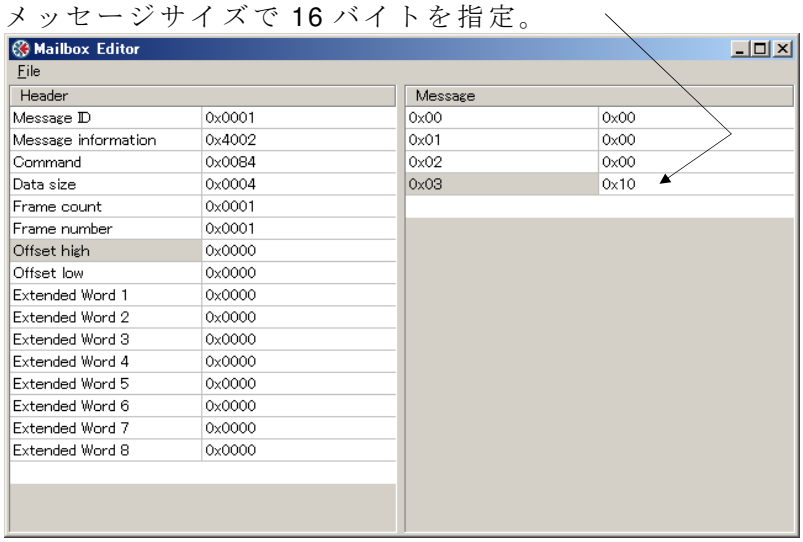


I/O サイズとして”IO Size In”を 0x100 バイトと指定します。

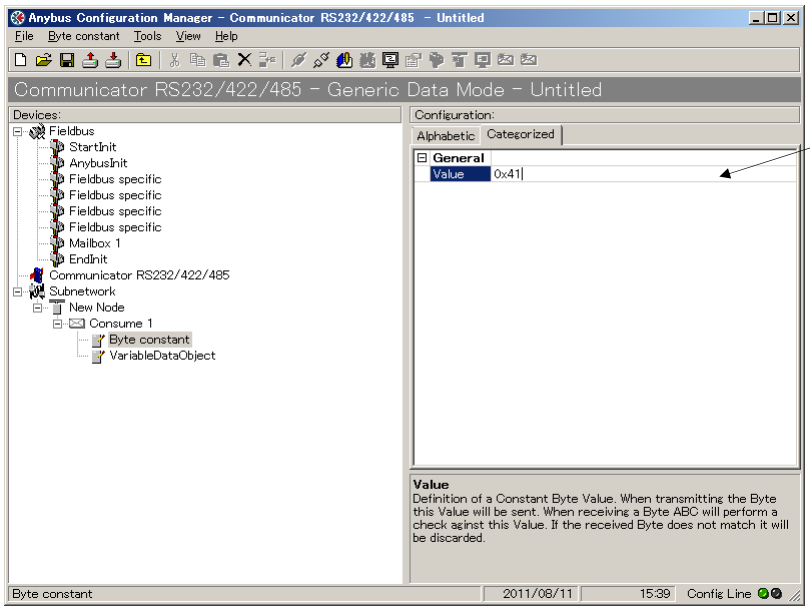


上記の設定により、0x00 番地から 0xFF 番地までが I/O データ領域となり、残りの 0x100 番地からがメッセイージデータ領域となります。

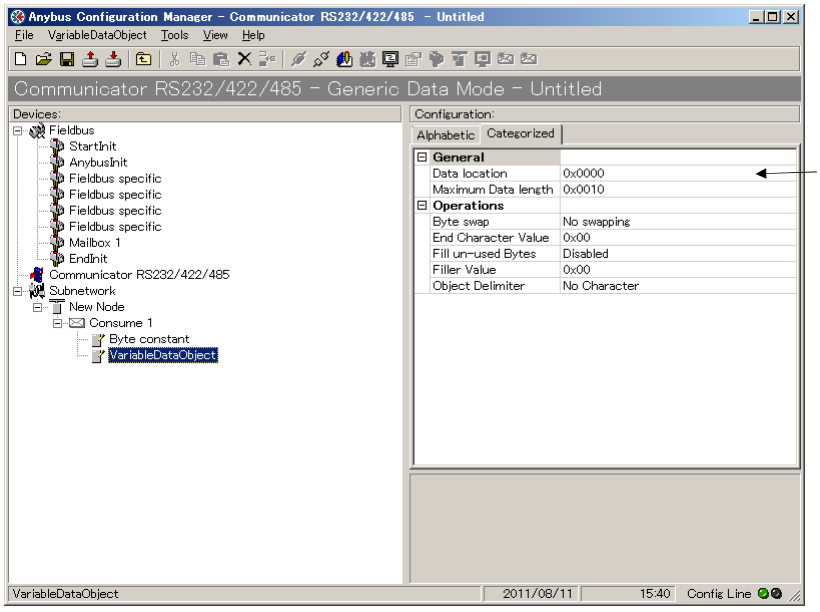
メッセージサイズで 16 バイトを指定。



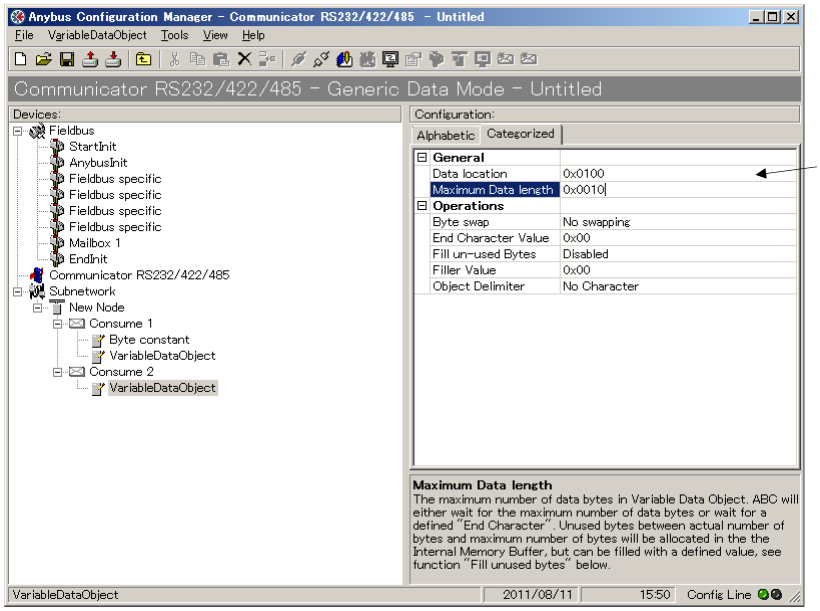
以後はシリアル側の設定を行います。先頭番地が'A'(0x41)の時に処理されるように Consume1 を設定します。



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。



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



Consume2 はすべてのデーターが 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

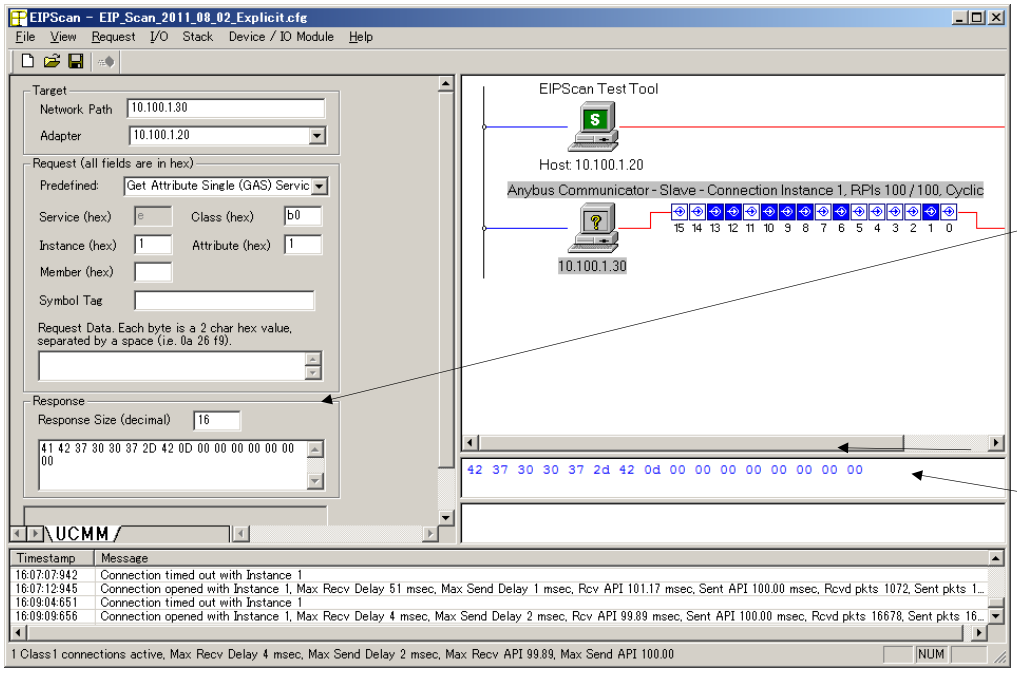
Originator -> Target: 0

Target -> Originator: 16 <=I/O データーサイズ

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 データーとメッセージデーターの両方が EIP Scan に取り込まれています。



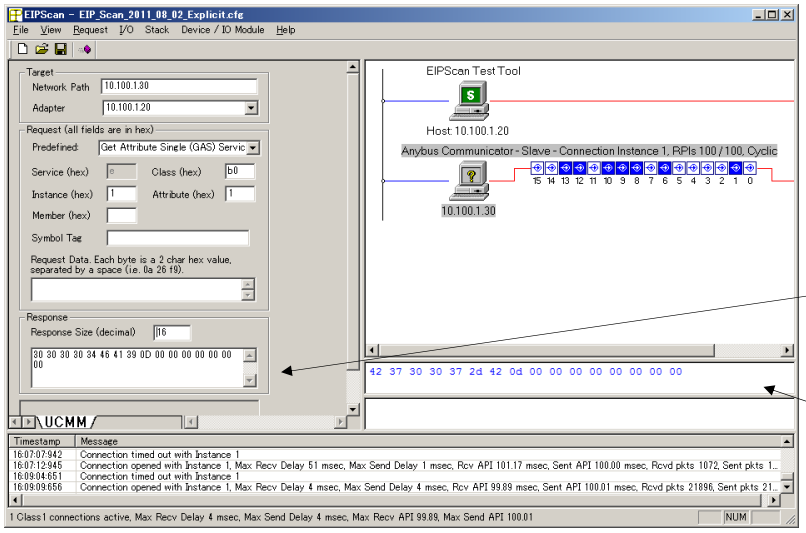
メッセージデーター（”AB7007-B”+0x0D）が取り込まれています

I/O データー（”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ーの先頭の’A’(0x41)は、“Byte Constant”として指定されている為に I/O データーとしては認識されない。 よって、I/O データーとしては次のデーター’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