

# Anybus Communicator (ABC)

## 各種設定方法 (1)

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)

<b>EVOLUTION OF THE DOCUMENT .....</b>	<b>3</b>
<b>1. サブネットワーク（シリアル）側からくる不要なデータを転送しない .....</b>	<b>4</b>
<b>2. I/O データーびメッセージデーターのメモリー割り当て .....</b>	<b>5</b>
2.1. 全メモリー領域(入力領域)を I/O データーとして使用 .....	6
2.2. I/O データー(入力領域)とメッセージ領域(入力領域)の両方を使用 .....	8
2.3. 全メモリー領域(入力領域)をメッセージデーターとして使用 .....	10
2.4. 全メモリー領域(出力領域)を I/O データーとして使用 .....	11
2.5. I/O データー(出力領域)とメッセージ領域(出力領域)の両方を使用 .....	11
2.6. 全メモリー領域(出力領域)をメッセージデーターとして使用 .....	11

**EVOLUTION OF THE DOCUMENT**

---

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

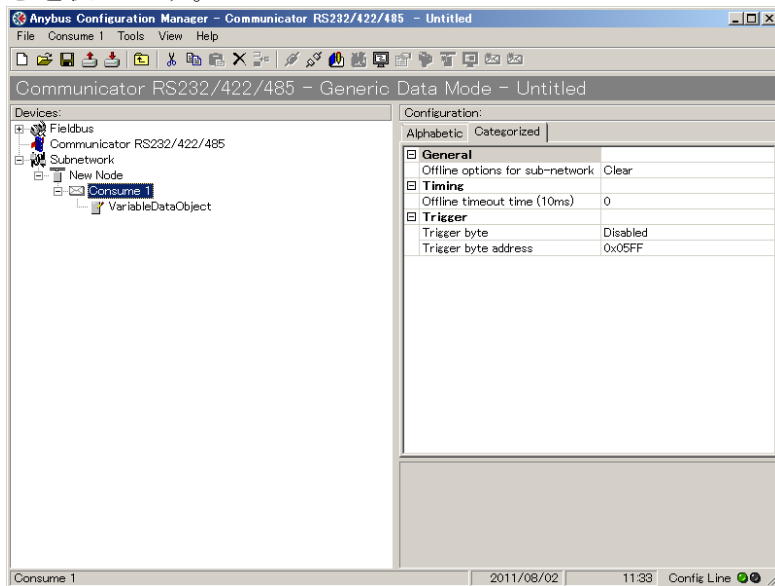
---

---

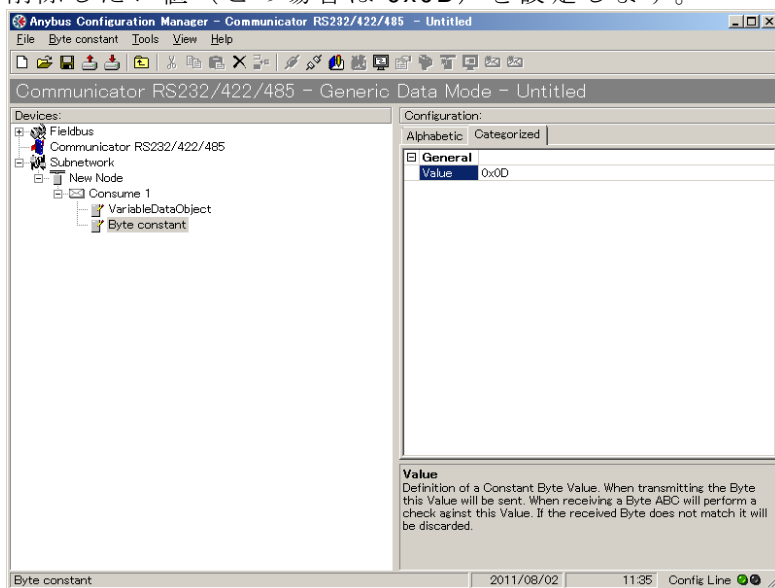
This document contains: 11 pages.

## 1. サブネットワーク（シリアル）側からくる不要なデータを転送しない

例） 最後のリターンコードを削除したい場合、Consume1 上で右クリック、"Add Byte, Constant" を選択します。



削除したい値（この場合は 0x0D）を設定します。



## 2. I/O データとメッセージデータのメモリ割り当て

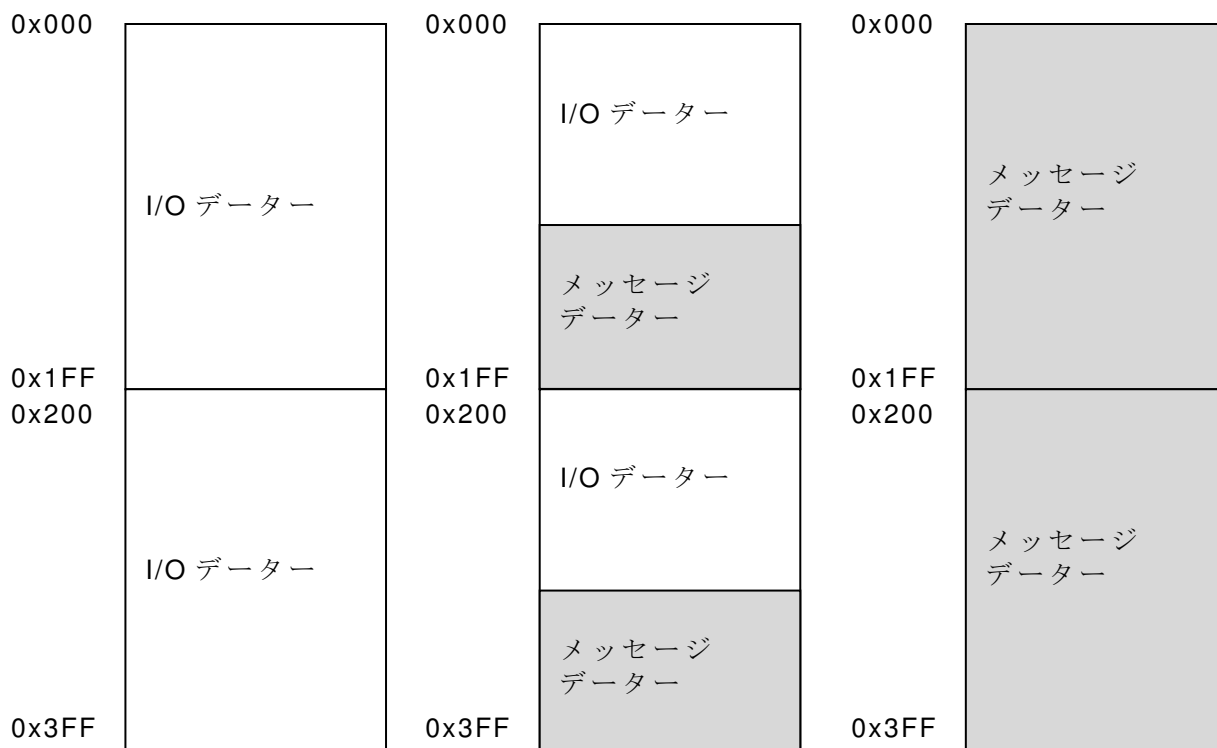
メッセージデータ通信を行う場合は、CIP の仕様を十分理解された後で御使用下さい。以降の説明は CIP に関しての仕様を理解された方を対象として説明致します。

ABC においてメッセージデータ通信を行う場合は、ABC のメモリ内にメッセージデータ通信用のメモリをマッピングする必要があります。

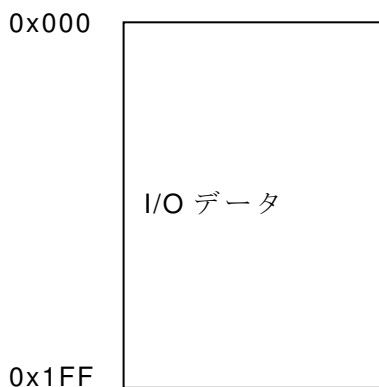
ABC 内にあるメモリサイズは、入力領域 (IO Size In 領域(シリアル側から ABC に入力されるデータを取込む為の領域)) として 0x000-0x1FF (512 バイト)、又、出力領域 (IO Size Out 領域(ABC よりシリアル側へデータを送る為の領域)) として 0x200-0x3FF(512 バイト)存在します。



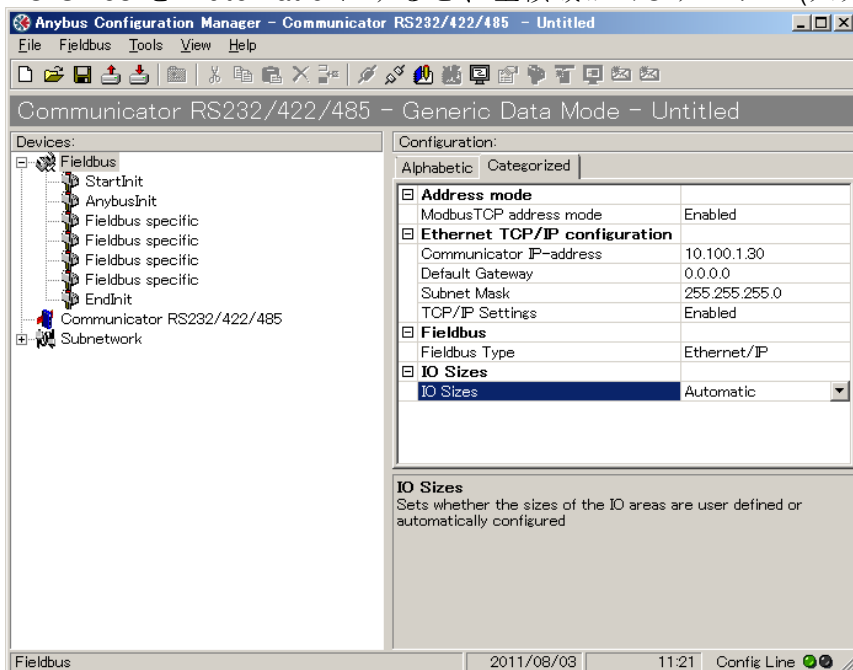
そのメモリを I/O データ領域とメッセージデータ領域で分けて使用します。



## 2.1. 全メモリー領域(入力領域)を I/O データーとして使用



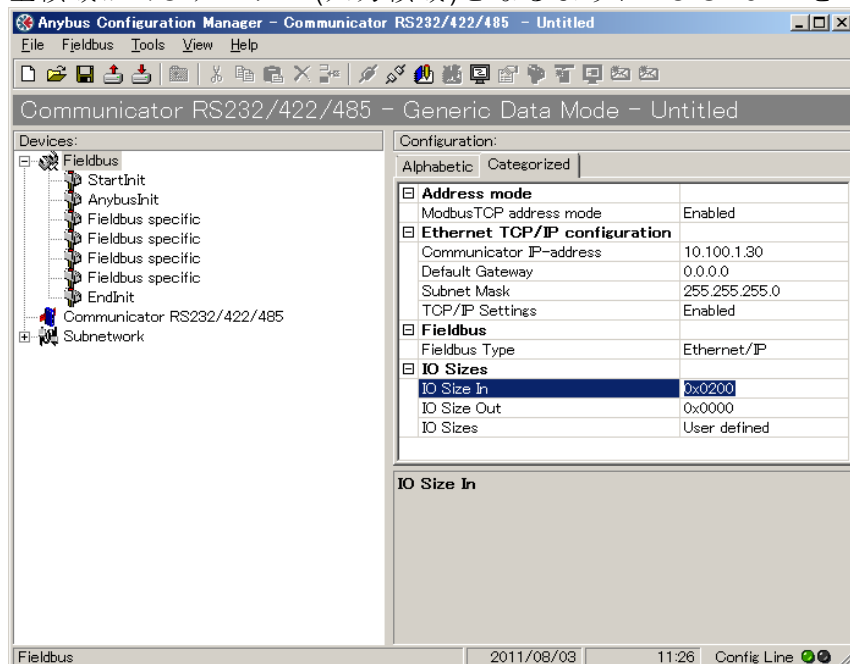
“IO Sizes”を”Automatic”にすると、全領域が I/O データー(入力領域)となります。



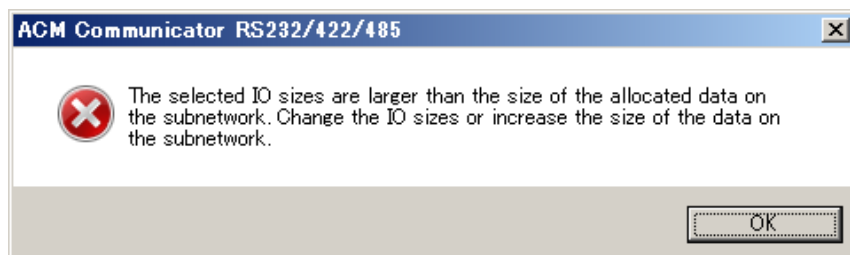
全領域を I/O データーとして使用する場合は、“IO Sizes”を”Automatic”で御使用下さい。

“IO Sizes”を“User defined”にして使用する場合は注意が必要です(本使用方法是参考までの説明です)。

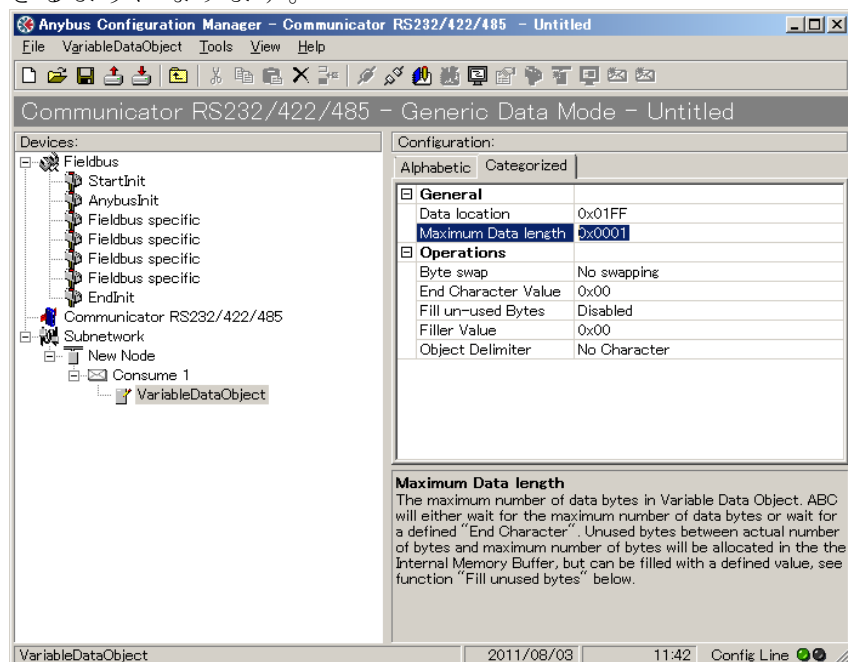
全領域が I/O データ(入力領域)となるように“IO Size In”を 0x200 と指定します。



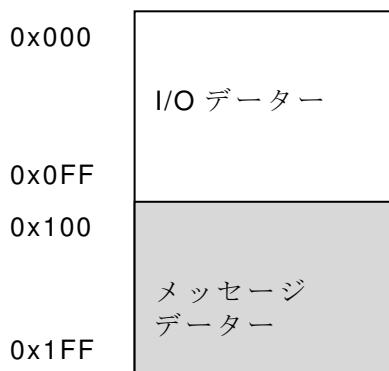
この場合、サブネットワーク側の“Data location”及び“Maximum Data Length”の指定で 0x200 以外の値 (0x1FF 以下の値) を設定した場合は、以下のエラーが出力されます。



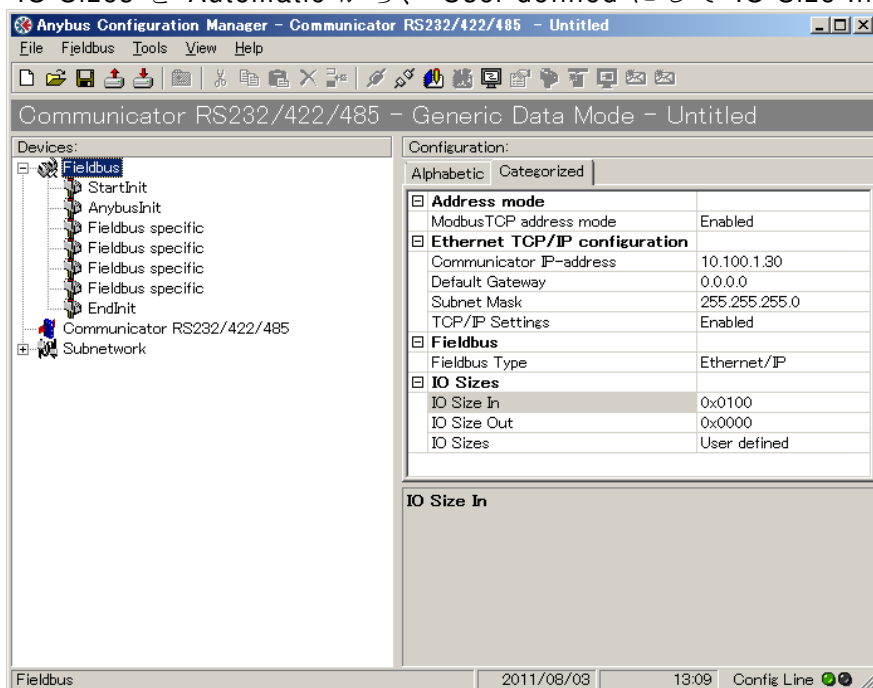
この回避策として、全領域を I/O データとして使用することを明示する為に、全領域の最後のデータ (1 バイト) を使用することを明示し、他の定義 (Consumee2,3,,) を行えば使用できるようになります。



## 2.2. I/O データー(入力領域)とメッセージ領域(入力領域)の両方を使用

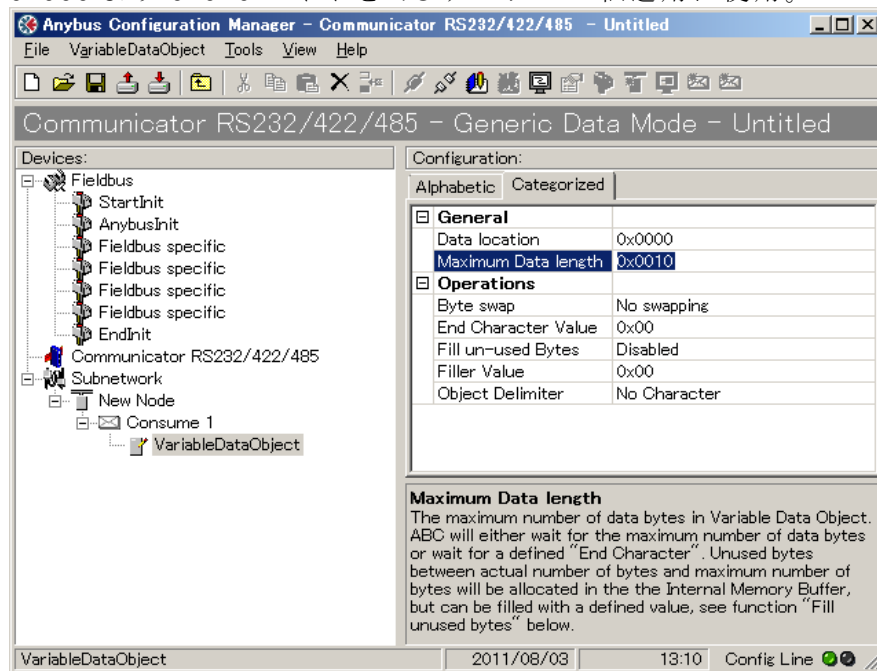


“IO Sizes”を“Automatic”から、“User defined”にして“IO Size In”を 0x0100 に設定する。

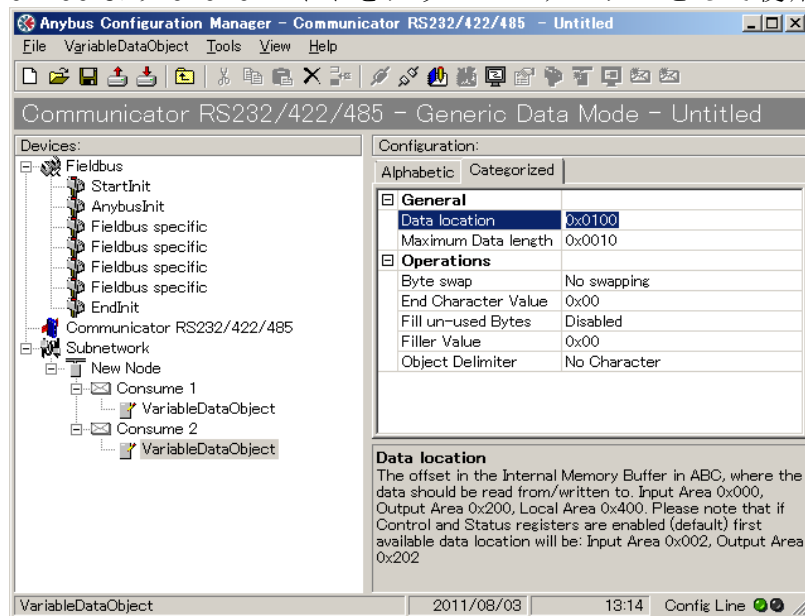




サブネットワーク側の”Data location”及び”Maximum Data Length”を指定、今回の例の場合は 0x000 より 0x010 バイトを I/O データの転送用に使用。



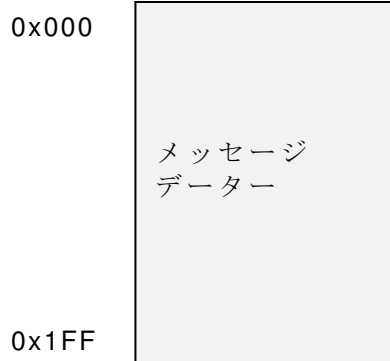
サブネットワーク側の”Data location”及び”Maximum Data Length”を指定、今回の例の場合は 0x100 より 0x010 バイトをメッセージデータとして使用。



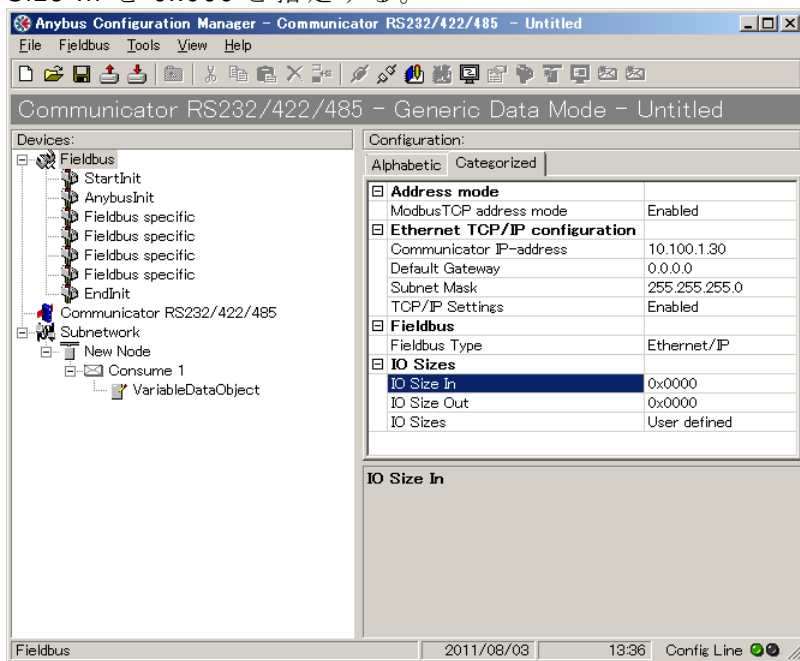
本例では、”IO Size In”として 0x100 を指定している為（IO サイズとして 0-0x0FF まで取得）、アドレス 0x100 からはメッセージデータとして扱われます。その為に、メッセージ領域の定義が必要となり、”Consume2”の定義を加えています。

I/O データ（入力領域）を 0x000 – 0x0FF、メッセージ領域（入力領域）を 0x100-0x1FF として使用。この場合、取りこまれたデータは I/O データとして 0x000 から、メッセージデータとして 0x100 から入ります。

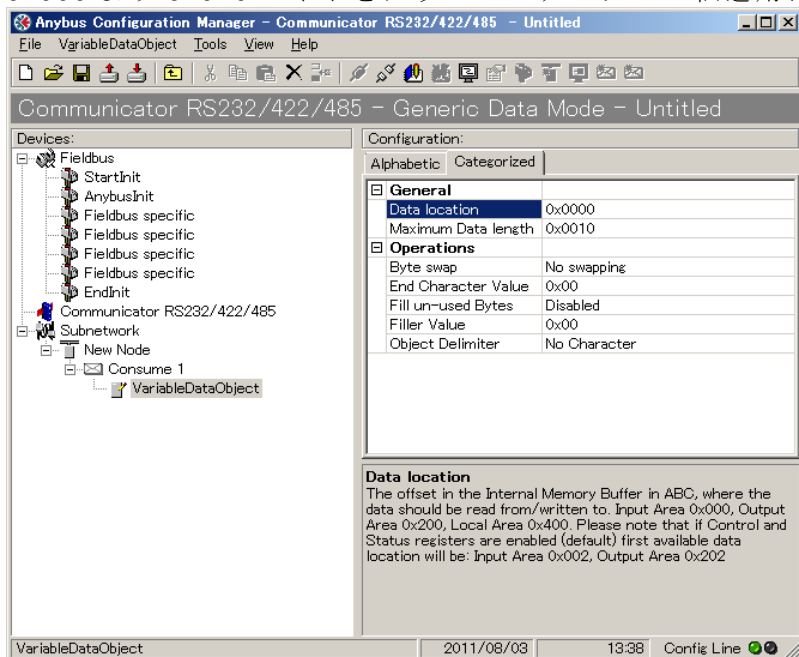
## 2.3. 全メモリー領域(入力領域)をメッセージデーターとして使用



“IO Sizes”を”User defined”にして、全領域がメッセージデーター(入力領域)となるように”IO Size In”を 0x000 と指定する。



サブネットワーク側の”Data location”及び”Maximum Data Length”を指定、今回の例の場合は 0x000 より 0x010 バイトをメッセージデーターの転送用に使用。



## 2.4. 全メモリー領域(出力領域)を I/O データーとして使用

“IO Sizes”を”Automatic”にすると、全領域が I/O データー（出力領域）となります。

## 2.5. I/O データー(出力領域)とメッセージ領域(出力領域)の両方を使用

“IO Sizes”を”Automatic”から、”User defined”にして”IO Size Out”を 0x0100 に設定する。サブネットワーク側の設定は前項 “I/O データー（入力領域）を 0x000-0x1FF、メッセージ領域（入力領域）を 0x100-0x1FF として使用 “を参照。

## 2.6. 全メモリー領域(出力領域)をメッセージデーターとして使用

“IO Sizes”を”User defined”にして、全領域がメッセージデーター(出力領域)となるように”IO Size Out”を 0x000 と指定する。