

**M2M/IoTソリューション CONPROSYS
FITプロトコル通信 サンプルタスク**

**2018年03月06日
株式会社コンテック**

必要機材

本サンプルを利用するに当り、最低限必要な機材を以下に示します。
動作確認や信号状況を変化させるためのスイッチ・センサ類は適宜ご用意下さい。

品名	型式	必要数	メーカー
M2Mコントローラ	CPS-MC341-ADSC1-111	2	CONTEC
ノートPC	-※1	1	
LANケーブル	-※2	2	

※1 : Google Chrome、Firefox、Internet Explorer11等が動作するPCをご利用下さい。

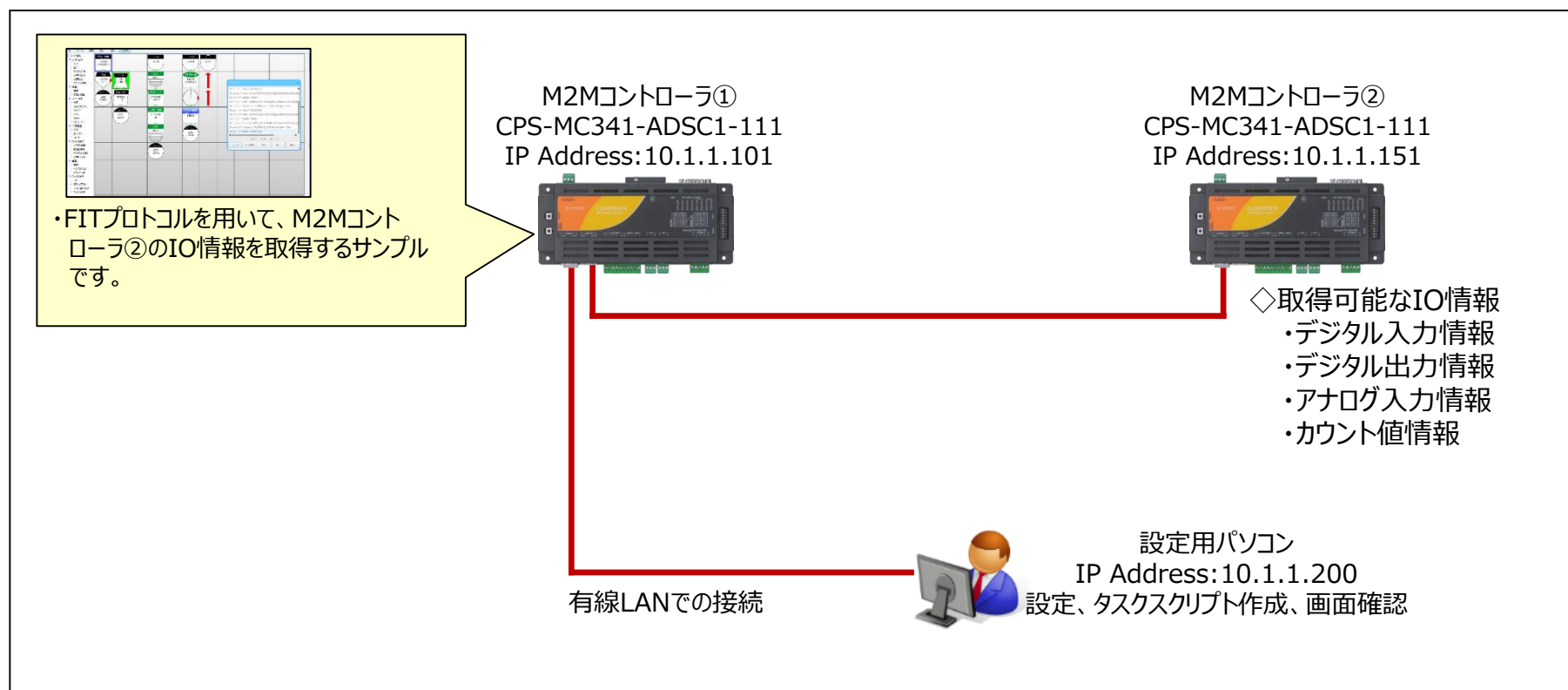
※2 : M2Mコントローラ本体は2つのLANポートを備えますので、HUBとしてご利用下さい。

その他のネットワーク機器を接続するような場合は、適宜HUBやケーブルをご用意下さい。

サンプル概要

本サンプルは2台のCONPROSYSをLANケーブルで接続し、FITプロトコルを利用してIO情報を取得するサンプルプログラムです。FITプロトコルを用いてデータを取得するサンプルタスクと、取得したデータを表示するサンプルモニタリング画面を用意しております。

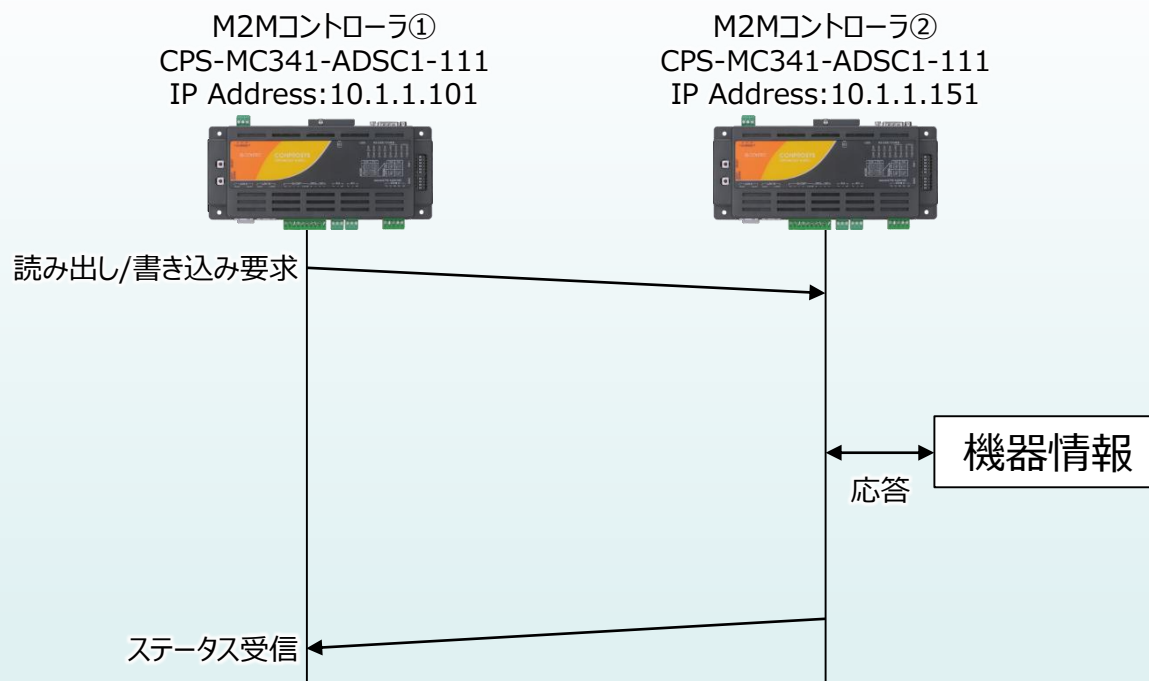
本サンプルを利用する場合のネットワーク構成を以下に示します。



FITプロトコル概要

F&eITプロトコル(以降FITプロトコル)はUDP/IPを用いた弊社独自の通信プロトコルで、主にリモートIO情報を取得することを目的にしたプロトコルです。本プロトコルは『 CONPROSYS®シリーズ 』の他、『 F&eIT® 省配線リモートI/Oシステム 』にも搭載されています。

通信イメージ



**FITプロトコルでは、UDPポート：20487を使用します。
また、バイトオーダーはリトルエンディアンです。**

サンプルタスク使用方法

- ① CONPROSYSのタスク編集画面より[ファイル]－[ローカルディスクから開く] を選択します。
- ② ダウンロードしたファイルを展開したフォルダから「FIT_Sample.dat」を選択し、[開く]を選択します。
- ③ CONPROSYSのタスク編集画面より[ファイル]－[タスクを保存...]を選択し任意のファイル名：Task0～9の何れかで保存します。
- ④ [設定]－[リンク設定...]を選択します。
- ⑤ 「リンク設定」画面の [link-0] の [詳細] を選択します。
- ⑥ 本資料P.4の内容にしたがって設定します。右図を参考下さい。
 - ・コネクションモード : アクティブモード
 - ・接続先ホスト : 10.1.1.151
 - ・プロトコル : UDP
 - ・ポート番号 : 20487
 - ・通信タイプ : 可変長
 - ・設定値 :
 - ・無通信タイムアップ(x100msec) : 1
 - ・リンクオープンモード : 常時オープン

*1：最終電文受信後、設定値(×100msec)の間のみ次のパケット受信を待ちます。
パケットが分割されるような際にご利用下さい。

項目	設定値
接続タイプ	Ethernet
コネクションモード	アクティブモード
接続先ホスト	10.1.1.151
プロトコル	UDP
ポート番号	20487
通信タイプ	可変長
設定値	
無通信タイムアップ(x100msec)	1
リンクオープンモード	常時オープン

サンプルタスク動作

サンプルタスクを以下に示します。本ページでは①部分の説明を致します。



① 前処理～FITプロトコル通信

- 1秒間待機します。
- 受信バッファ(STAG00)をクリアします。
- FIT通信を実施します。FITプロトコル通信は、以下の条件で設定しています。

◇サンプルタスク『FITプロトコル』プロパティ情報



プロパティ	値
リンクNo	link-0
アドレス(HEX)	固定値
固定値(アドレス)	3a2900
FeITアクセス	読み込み
サイズ	固定値
固定値(サイズ)	48
バッファ	STAG00
次ステップ	下へ
→ X	0
↓ Y	2

設定したリンク設定番号を指定下さい。

FITプロトコルにおけるアクセスするアドレスを指定します。アクセスアドレスは、リファレンスマニュアル等を参照下さい。

指定したアドレスから読み出すサイズを指定します。アドレスマップはリファレンスマニュアル等を参照下さい。

受信電文をSTAG00に代入します。

◇CPS-MC341ADSC1-111 FITプロトコルアドレス情報

◆ I/O空間

I/O情報(例)

- CPS-MC341-ADSC1-111
- CPS-MC341Q-ADSC1-111
- CPS-MC341G-ADSC1-110
- CPS-MC341-ADSC1-111
- CPS-MC341G-ADSC1-111

アドレス	サイズ		内容	補足
3A2900h~	1	R	DI-0ch	bit0, bit1, bit2, bit3
	1	RW	DO-0ch	
	14	R	dumy	
	2	R	AI-0ch	LSB(0-4095)
	2	R	AI-1ch	
	12	R	dumy	
	4	R	CNT-0ch	LSB(0-16777215)
	4	R	CNT-1ch	

サンプルタスクを以下に示します。本ページでは②部分の説明を致します。



② 受信内容確認～通信異常判定

- I. 受信バッファ(STAG00)のデータサイズを確認します。データサイズ確認は、『数値に変換』アイコンを用います。データ取得に失敗している場合、STAG00内はnullとなり、データサイズを取得した場合は0になります。

◇サンプルタスク『数値に変換』プロパティ情報



プロパティ	値	
演算結果格納先=	TAG00	→ 変換結果の代入先を選択します。
実行	データサイズ取得	→ データサイズ取得を選択します。
文字列	STAG00	→ データサイズを取得する文字列変数を選択します。
次ステップ	下へ	
→ X	0	
↓ Y	3	

- II. I.の処理にて、データサイズを取得した後判定を用いて正常・異常を判断します。
受信データサイズ(TAG0)の値が0より大きければ正常、それ以外は異常とします。
- III. 正常の場合は、通信状態(TAG10)に1を代入し、③の処理へジャンプします。
異常の場合は通信状態(TAG10)に0を代入し、先頭に戻ります。

サンプルタスクを以下に示します。本ページでは③部分の説明を致します。



③ データ抽出処理

受信バッファ内の電文から必要なデータを抜き出していきます。
データの抽出には、『数値に変換』アイコンを用います。取得するサイズに応じてプロパティを変更しています。

◇DI/DO情報取得時 (I・II)

プロパティ	値
演算結果格納先=	TAG01
実行	1ByteバイナリからSHORT
文字列	STAG00
先頭からの位置	固定値
固定値 (先頭からの位置)	DI : 0、DO : 1

◇CNT情報取得時 (III・IV)

プロパティ	値
演算結果格納先=	TAG05
実行	4ByteバイナリからLONG
文字列	STAG00
先頭からの位置	固定値
固定値 (先頭からの位置)	CNT00:32、CNT01:36
バイト順	リトルエンディアン

◇AI情報取得時 (III・IV)

プロパティ	値
演算結果格納先=	TAG03
実行	2ByteバイナリからSHORT
文字列	STAG00
先頭からの位置	固定値
固定値 (先頭からの位置)	AI00:16、AI01:18
バイト順	リトルエンディアン

先頭からの位置及び、取得サイズはアドレスマップを参照下さい。

◇CPS-MC341ADSC1-111 FITプロトコルアドレス情報

アドレス	サイズ		内容	補足
3A2900h~	1	R	DI-0ch	bit0、bit1、bit2、bit3
	1	RW	DO-0ch	
	14	R	dumy	
	2	R	AI-0ch	LSB(0-4095)
	2	R	AI-1ch	
	12	R	dumy	
	4	R	CNT-0ch	LSB(0-16777215)
	4	R	CNT-1ch	
	8	R	dumy	

TAG割付

本サンプルで用いているTAGの割付情報を以下に示します。
拡張及び連携する際の参考としてご使用下さい。

TAG番号	項目名
TAG00	受信データサイズ
TAG01	DI情報
TAG02	DO情報
TAG03	AI00情報
TAG04	AI01情報
TAG05	CNT00情報
TAG06	CNT01情報
TAG07	予備
TAG08	予備
TAG09	予備
TAG10	通信状態

STAG番号	項目名
STAG00	受信バッファ
STAG01	予備
STAG02	予備
STAG03	予備
STAG04	予備
STAG05	予備
STAG06	予備
STAG07	予備
STAG08	予備
STAG09	予備
STAG10	予備

サンプルモニタリング画面使用方法

- ① CONPROSYSのモニタリング編集画面より[ファイル]–[ローカルディスクから開く] を選択します。
- ② ダウンロードしたファイルを展開したフォルダから「FIT_Sample.page」を選択し、[開く]を選択します。
- ③ [ファイル]–[名前をつけて保存...]を選択し任意のファイル名で保存します。

HMIの画面イメージ

モニタリング画面 データ確認サンプル

FITプロトコル通信データを確認するためのサンプルモニタリング画面です。

通信状態を表示します。
通信異常時には、下記表示となります。

通信異常

割付TAG : TAG10

正常通信

取得情報

受信バイト長	48	割付TAG : TAG00
DI0-3状態	4	割付TAG : TAG01
DO0-1状態	0	割付TAG : TAG02
AI0計測値	0	割付TAG : TAG03
AI1計測値	0	割付TAG : TAG04
CNT0計測値	18	割付TAG : TAG05
CNT1計測値	0	割付TAG : TAG06

通信より取得した情報を表示します。なお、本サンプルでは通信異常時には前回通信した値が残ります。通信異常時には0クリアする場合は、本資料6ページの通信異常時判定後にクリア処理等を追加下さい。

FITプロトコル通信注意事項

▼ FITプロトコル 受信電文表示に関して

FITプロトコルで取得した電文を、デバッグ画面、もしくはモニタリング画面で表示する時、データが何も入っていないように見えることがあります。

これは、FITプロトコルはバイナリ値を扱うため『0x00』が入ることがあるためです。『0x00』が含まれた文字列をデバッグ画面やモニタリング画面で表示しようとする時、文字コード上『0x00』が終端文字と解釈され、そこで文字列としては終了となります。

以下に例を示しますが、データ受信時に0x00が含まれているとそれ以降デバッグ画面等で表示されませんが、実際にデータとしては入っておりますので、受信確認には『数値に変換』のデータ長プロパティをご利用下さい。



『0x00 0x01 0x00 ...』
というデータを受信した場合(計48Byte)

デバッグ

```
Act:x=0 y=0<s_fit> link0 addr[3a2900] read size[48] buf[STAG000()]
```

```
Act:x=0 y=1<s_calc> STAG005[!=(= Str[STAG000()]) >>[]
```

```
Act:x=0 y=2<s_2num> TAG000=(method[GetLength]str[STAG000]) >>[0]
```

```
Act:x=0 y=3<s_2num> TAG000=(method[GetBinarySize]str[STAG000]) >>[48]
```

```
Act:x=0 y=4<s_2num> TAG000=(method[1byte binary to short]str[STAG000]Offset[1]) >>[1]
```

デバッグ画面等でSTAGの中身を確認しようとしても、文字列表記時には、0x00が終端として扱われるためそれ以降は何も入っていないように見えてしまいます。
※今回の場合、先頭が0x00なので0やnullになります。

文字列ではなく、Byte長取得や、数値に変換等を用いるとデータが入っていることが確認できます。
⇒実際にデータが入っている事を確認できます。

