

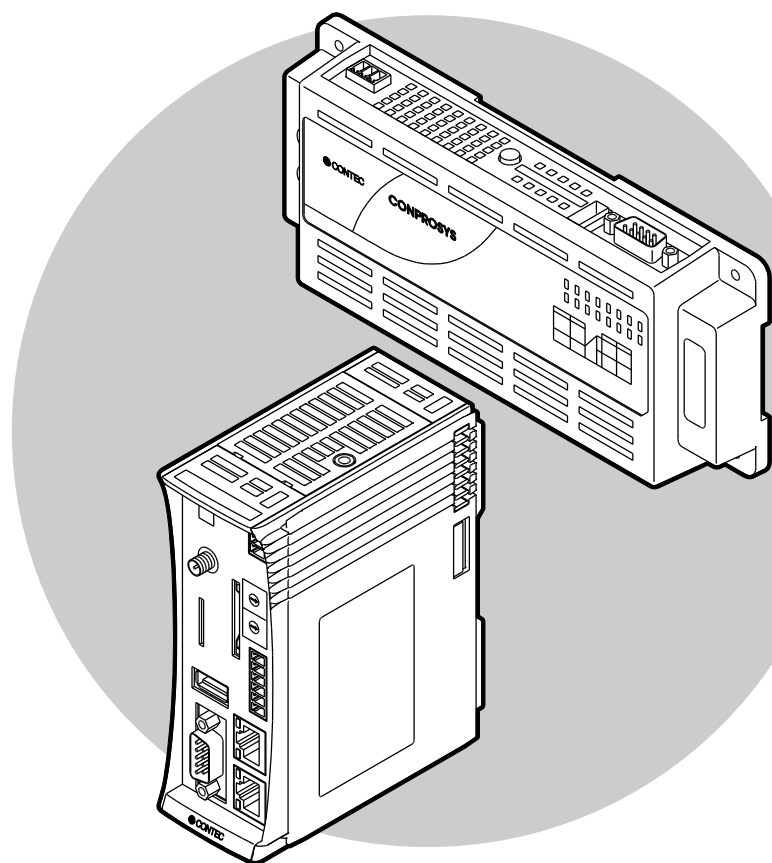
VTC スクリプト100選

(No.030)

CONPROSYS バイナリ通信サンプルタスク

目次

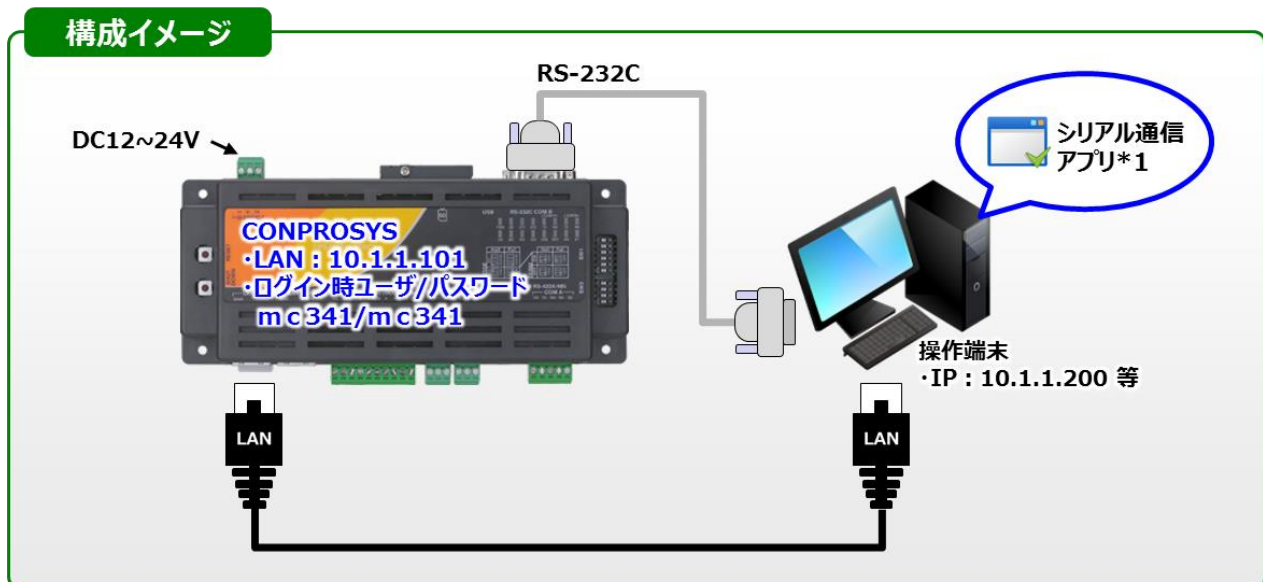
1. 概要	2
2. 使用機器一覧	3
3. 設定概要	4
4. サンプル使用方法	5
5. VTC解説	7
6. TAG定義	13
7. 参考情報	14



1. 概要

CONPROSYSとシリアル通信ソフトを利用し、バイナリ通信を実施するサンプルです。

- モニタリング画面から、ASCIIで入力された文字(0:0x30、1:0x31、…)をバイナリ値(0:0x00、1:0x01、…)に変換し、RS-232Cで送信します。
例1：『01(0x30 0x31)』を入力し、『0x01』に変換し、送信。
- RS-232Cで受信したバイナリ値(0:0x00、1:0x01、…)を、ASCII文字(0:0x30、1:0x31、…)に変換し、モニタリング画面に表示します。
例2：『0x01』を受信し、『01(0x30 0x31)』に変換し、表示。



*1 : シリアル通信にて、バイナリの送受信を行えるソフトウェアをご用意下さい。

2. 使用機器一覧

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 ※1	1	CONTEC
ノートPC	- ※2	1	-
LANケーブル	-	1	-
シリアル通信ソフトウェア	- ※3	1	-

※1：本型式以外でも、CONPROSYSシリーズなら動作致します。

※2：Google Chrome、Firefox、Internet Exproler11等が動作するPCをご利用下さい。

※3：ノートPC上のシリアル通信をソフトウェア用いて確認下さい。

バイナリ通信をサポートしたソフトウェアを適宜ご用意下さい。

3. 設定概要

本サンプルをご利用いただく際の、全体的な流れを紹介いたします。

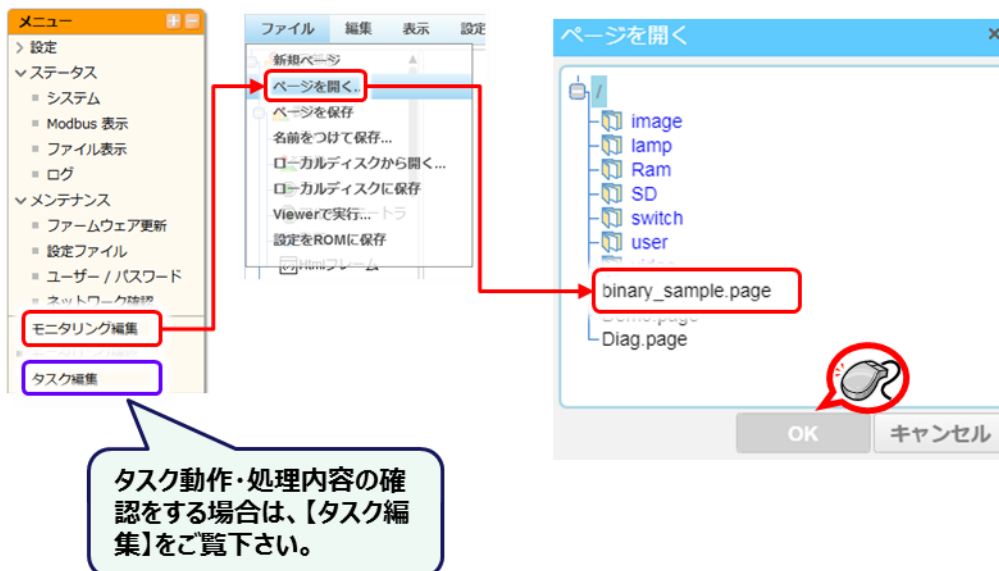
- 1 シリアル通信ソフトをインストール・PCネットワーク設定
お客様にて準備されたシリアル通信ソフトをインストールし、PCのIPを10.1.1.200に設定下さい。
※本設定の詳細説明は割愛致します。

- 2 サンプルファイル適応
ダウンロード頂いたサンプルファイルをお手元のコントローラに反映頂き、各種設定を確認下さい。
 - 2.1 タスク・モニタリング画面確認
各サンプルが反映されているか確認下さい。
確認完了後、コントローラにて設定保存及び再起動を実施下さい。
※後述の5ページ目を参照下さい。

- 3 動作確認
CONPROSYS再起動後、モニタリング画面にて、各種動作を確認下さい。
※後述の6ページ目以降を参照下さい。

4. サンプル使用方法

- CONPROSYSのメニュー画面：メンテナンス欄より、[設定ファイル]–[リストア] –[参照]をクリックします。
- ダウンロードしたファイルを展開したフォルダから「binary_sample.dat」を選択し、[更新]を選択します。
- メニュー画面の[終了]–[保存と再起動]を選択します。
- 再起動後、メニュー画面中の[モニタリング編集]にてサンプルモニタリング画面を表示下さい。



◇モニタリング画面 サンプル画面

ASCII文字からバイナリ値への変換及び送信を実施するサンプルと、バイナリ値で受信した値をASCII文字に変換し表示するサンプルです。

- ①送信サンプルを利用する場合は、文字を入力後、変換・送信ボタンを押下下さい。
- ②受信サンプルは常時起動しています。電文受信時に都度反映されます。

CONPROSYSバイナリ通信デバッグサンプル

・ CONPROSYSにてバイナリ通信を行うサンプルです。
 ・ 以下フォームにてASCIIで入力した値を、バイナリに変換してlink-0(デフォルト:COMB)に送信します。
 ・ link-0(デフォルト:COMB)から受信したバイナリデータをASCIIに変換して表示します。

バイナリ送信サンプル

文字入力フォーム: STAG00
※ 01F00E のような形で入力下さい。

バイナリ変換文字: STAG01
※ 基本的に表示されません。

変換確認文字: STAG15

入力文字長: 20 TAG00

※入力後クリック下さい。TAG99

バイナリ受信サンプル

受信電文(変換後): STAG20

受信電文(カンマ区切り): STAG25

受信電文(0x表記): STAG30

受信バイト長: 8 TAG05

入力されたASCII文字をバイナリ値に変換し、送信致します。TASK:0にて各種処理を実施しています。

各パーツに使用されている変数を示します。

RS-232Cポートにて受信したバイナリ値をASCII文字に変換し表示します。TASK:1にて各種処理を実施しています。

◇サンプルタスク説明：各種設定

・本サンプルにて実施してある設定を示します。実際にご使用される環境にあわせ、適宜変更下さい。

①リンク設定

- ・タスクで使用するLink設定は、Link-0に以下の設定をしています。
- ・実際にご利用になる機器に応じて、適宜ご設定下さい。

link-0 RS-232C通信を実施する設定です。×

接続タイプ:

ポート:

ボーレート:

データビット長:

パリティ:

ストップビット:

フロー制御:

全二重通信:

5. VTC解説

◇サンプルタスク説明 : Task:0 概要

- ・本サンプルでは、Task:0、Task:1を使用しています。Task:0の概要を以下に示します。
- ・Task:0はモニタリング画面で入力された文字を、バイナリ値に変換して送信するタスクです。

処理イメージ

入力文字 : 01 00 FF

③:抜き出し
③':2ループ目(以後同処理)
③":3ループ目(以後同処理)

演算用変数(文字:ASCII)
01(0x30,0x31)

変数(文字:Binary)
0x01

出力用変数(文字:Binary)
0x01(ループ毎に追記)

④:変換

⑤:追記

変換終了?
No ③^
※ループ回数+1

電文送信

⑥:変換完了後送信

『0x01 0x00 0xFF』と
いうバイナリ値を送信

①モニタリング画面中の実行ボタン押下を待機します。

↓

②ボタンクリック後、入力された文字列長をチェックします。

↓

③入力された文字の先頭2文字を抜き取ります。(ループ回数だけオフセットが掛かります。)

↓

④抜き出した文字を、バイナリ値に変換します。

↓

⑤変換したバイナリ値を出力用変数に追記し、ループ回数を+1します。

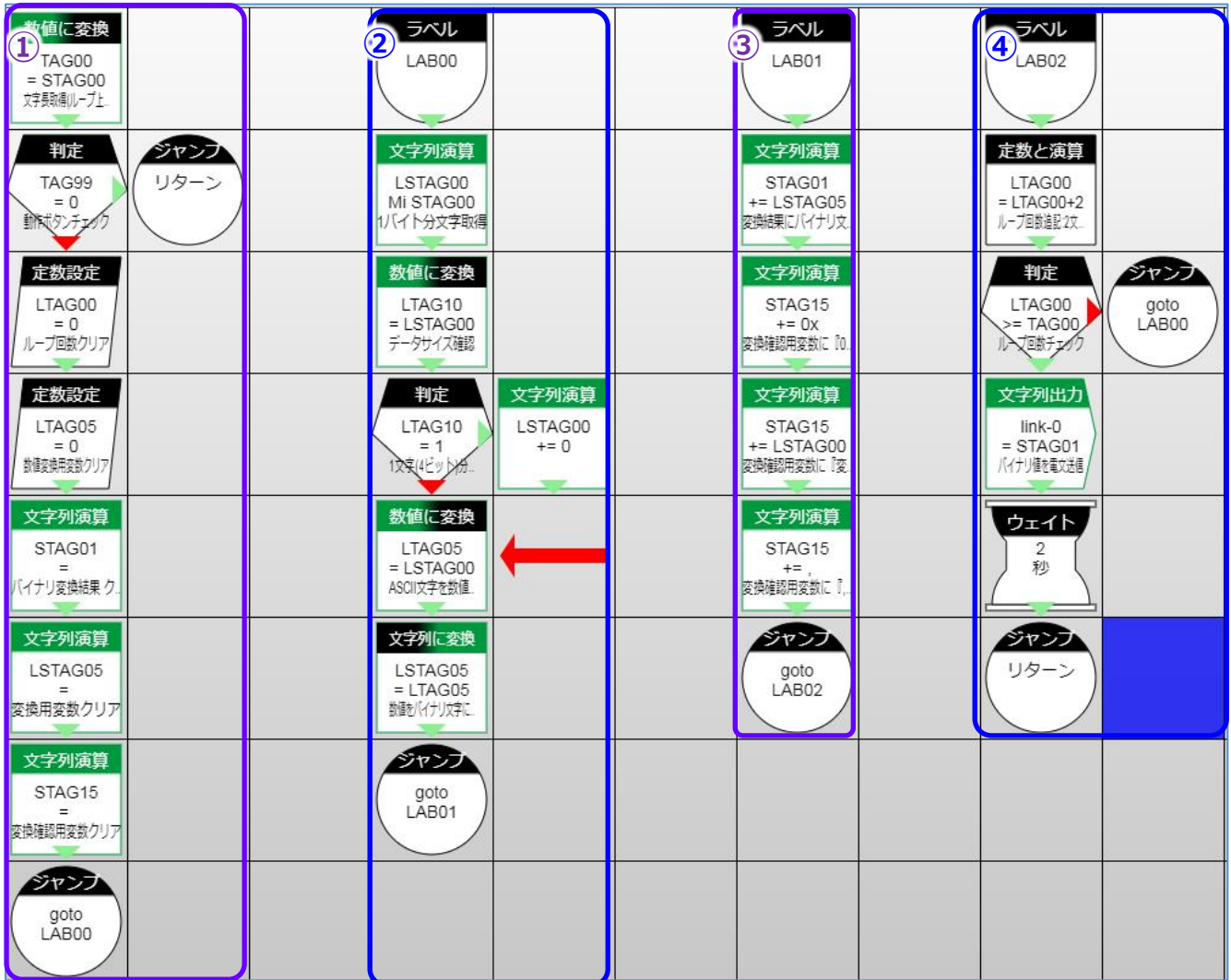
↓

⑥入力文字列長まで変換 (③~⑤)を繰り返し、完了後は送信します。

◇ サンプルタスク説明 : Task:0 概要

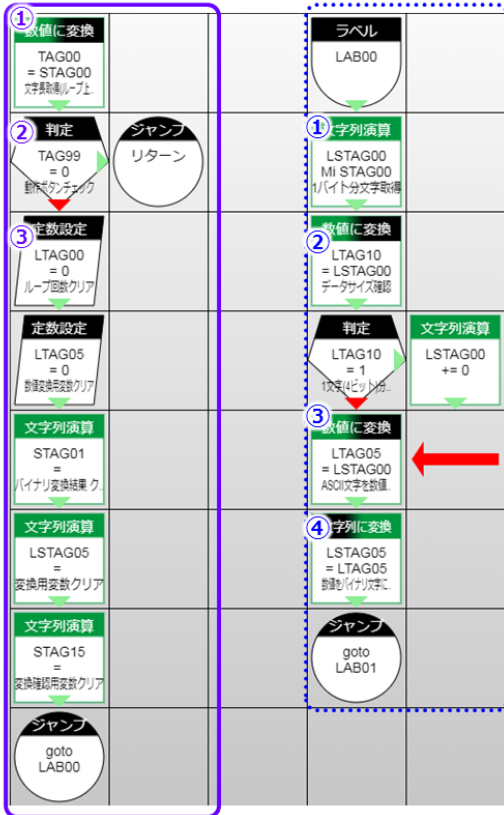
・ Task:0の全体を以下に示し、概要を説明します。

- ① モニタリング画面の送信ボタン押下を待機し、押下された場合は各変数を初期化して次の処理へ移ります。
- ② 入力された文字から1バイト分(2文字分)を抜き出し、バイナリ値に変換します。ループ回数毎に抜き出す位置をシフトしていきます。
- ③ 変換したバイナリ値を送信用変数に追記します。合わせて表示用変数にも追記します。
- ④ ループ回数を追加しループ回数が入力された文字列長を超えているかチェックします。超えている場合は Link-0 へ出力し、超えていない場合は②へ戻ります。



◇ サンプルタスク説明 : Task:0 詳細1

・ Task:0の前半部分を説明します。



実行ボタンの押下を待機し、押下後各変数をクリアする処理です。

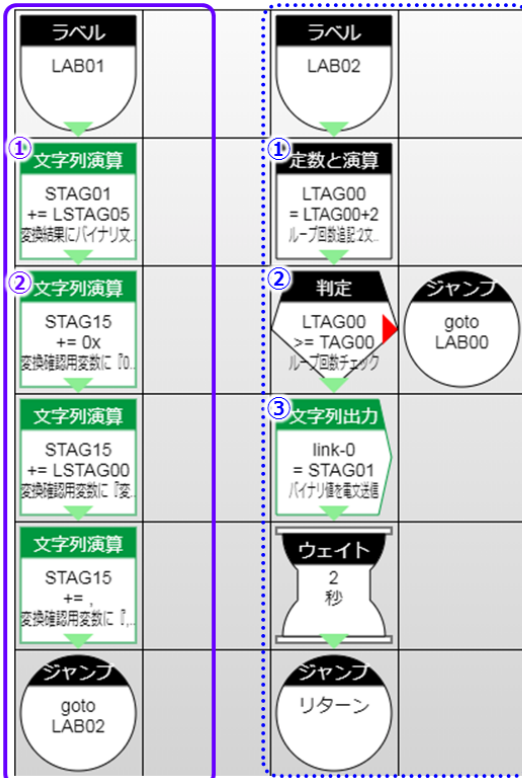
- ① モニタリング画面の入力フォーム値の文字列長を取得します。
- ② 実行ボタンが押下されるまで待機します。
- ③ 計算用の各変数をクリアします。

入力文字の先頭から1バイト分(2文字)を抜き出していく処理です。

- ① 文字列演算アイコンにて、入力文字の先頭2文字を抜き出します。(ループ毎にオフセット)
- ② 抜き出した文字の文字列長が1バイト分あるかの確認を行います。
⇒4ビット分の場合は、末尾に『0』を追加 ※『 0xF』という状態は不自然なため、『 0xF0』と変更します。
- ③ 抜き出した文字を、数値に変換
- ④ 変換した数値からバイナリ値を作成

◇ サンプルタスク説明 : Task:0 詳細2

・ Task:0の後半部分を説明します。



作成したバイナリ値の追記と、表示用の文字列の作成です。

- ① 作成したバイナリ値を、出力用変数 (STAG01)に追記します。
- ② 表示用変数(STAG15)も作成します。
※バイナリ値のままではモニタリング画面に表示されないため。

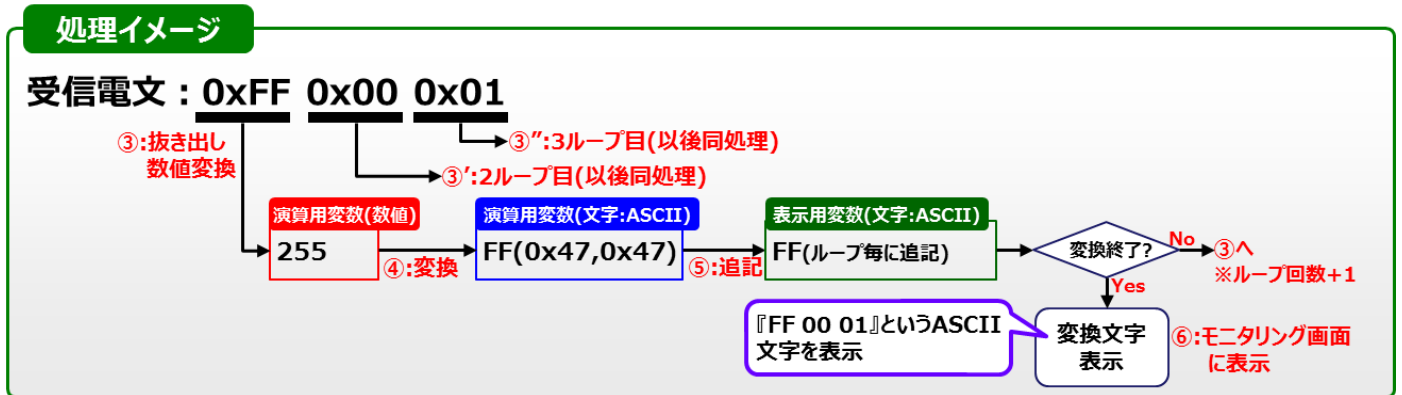
表示用変数は、『0xAA,0xBB...』という形で追記していきます。

ループ回数を加算した後に、回数チェックし、電文送信を実施します。

- ① ループ回数として+2します。
※1バイトずつ処理するため、2文字分をカウント致します。
- ② ループ回数が文字列長を超えているかチェック
- ③ 出力変数(STAG01)を送信します。

◇ サンプルタスク説明 : Task:1 概要

- ・本サンプルでは、Task:0、Task:1を使用しています。Task:1の概要を以下に示します。
- ・Task:1は受信したバイナリ値をASCII文字に変換し、表示するタスクです。

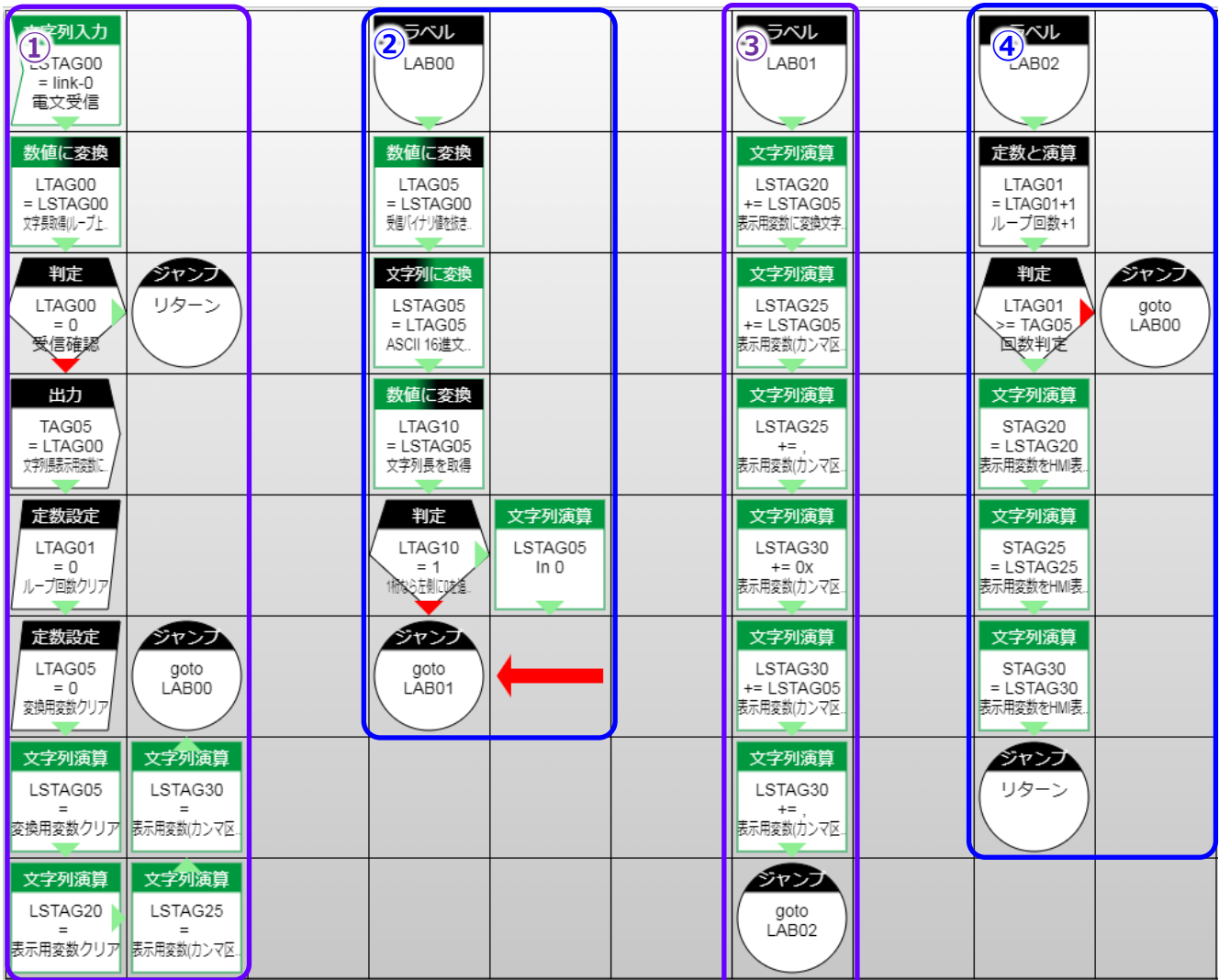


- ① 電文を受信するまで待機します。
- ↓
- ② 電文受信後、バイト長をチェックします。
- ↓
- ③ 受信した電文の先頭1バイトを抜き出し、数値に変換します。
(ループ回数だけオフセットが掛かります。)
- ↓
- ④ 抜き出したバイナリ値をASCII文字に変換します。
- ↓
- ⑤ 変換したASCII文字を表示用変数に追記し、ループ回数を+1します。
- ↓
- ⑥ 入力文字列長まで変換 (③~⑤)を繰り返し、完了後はモニタリング画面に表示します。

◇ サンプルタスク説明 : Task:1 概要

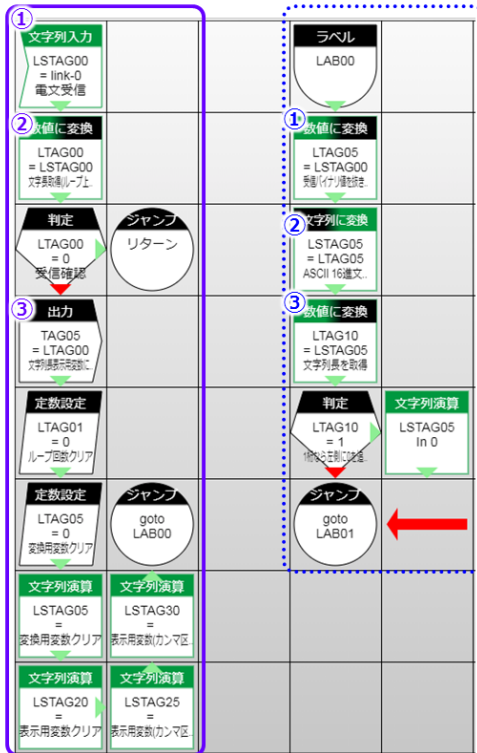
・ Task:1の全体を以下に示し、概要を説明します。

- ① Link-0からの電文受信を待機し、各変数を初期化して次の処理へ移ります。
- ② 受信した電文から1バイトを抜き出し、ASCII文字に変換します。ループ回数毎に抜き出す位置をシフトしていきます。
- ③ 変換したASCII文字を演算用変数に追記します。表示パターンは3種類用意しています。
- ④ ループ回数を追加し、ループ回数が受信したバイト長を超えているかチェックします。超えている場合は演算用変数を表示用変数に代入し、超えていない場合は②に戻ります。



◇ サンプルタスク説明 : Task:1 詳細1

・ Task:1の前半部分を説明します。



電文受信を待機し、受信確認後計算用変数をクリアします。

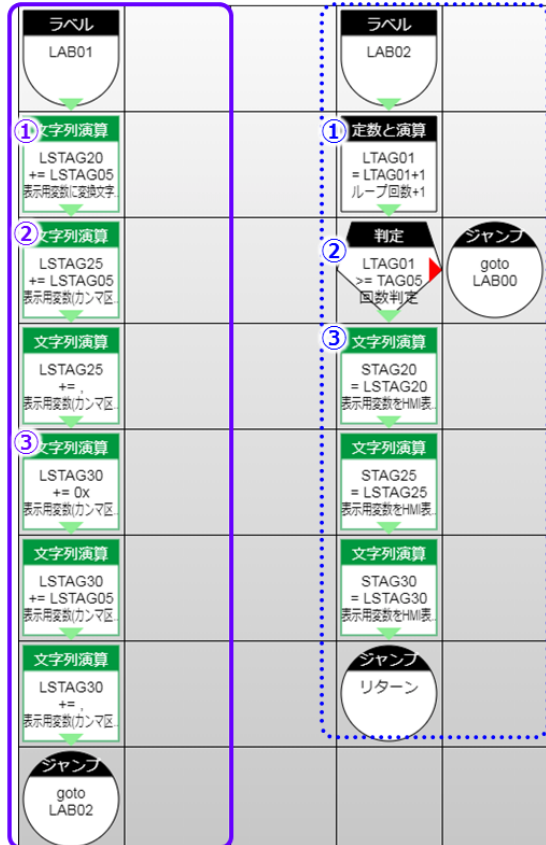
- 1 受信用変数(LSTAG00)に受信電文を代入します。
- 2 受信電文のバイト長を確認し、0で無ければ、次の処理に進みます。
- 3 各変数をクリアします。

入力文字の先頭から1バイト分を抜き出し、変換していき処理です。

- 1 『数値に変換』アイコンを利用し、受信電文の先頭1バイトを数値に変換します。(ループ毎にオフセット)
- 2 変換した数値をASCII文字に変換します。
- 3 変換結果、1文字の場合は、左に『0』を追記します。※『E』⇒『OE』とするため。

◇ サンプルタスク説明 : TASK:1 詳細2

・ Task:1の後半部分を説明します。



変換結果を表示用変数に追記します。
表示用変数は、以下の3パターン用意します。

パターン1 : STAG20/LSTAG20
AABB...
※そのまま追記して表示

パターン2 : STAG25/LSTAG25
AA, BB, ...
※カンマ区切りで表示

パターン3 : STAG30/LSTAG30
0xAA, 0xBB, ...
※0x表示を追加し、カンマ区切りで表示

- 1 パターン1表示変数に追記
- 2 パターン2表示変数に追記
- 3 パターン3表示変数に追記

ループ回数を加算した後に回数チェックし、表示用電文に代入します。

- 1 ループ回数を+1します。
- 2 ループ回数が文字列長を超えているかチェック。
- 3 各計算用変数を、表示用変数に代入します。
※LSTAG ⇒ STAG に代入。

6. TAG定義

◇使用TAG・STAG

・本サンプルで用いているTAG状況を以下に示します。(『…』は未使用を示します。)

TAG番号	TASK No.	内容
TAG00	TASK00	入力文字列長(STAG0文字数・ループ上限)
...		
TAG05	TASK01	受信文字列長(Link0からの受信サイズ)
...		
TAG15	TASK00	送信文字列確認用
...		
TAG99	TASK00	送信ボタン(HMI操作)

LTAG番号	TASK No.	内容
LTAG00	TASK00	ループ回数(ASCII⇒Binary変換処理)
	TASK01	受信データバイト長(ループ上限)
LTAG01	TASK01	ループ回数(Binary⇒ASCII変換処理)
...		
LTAG05	TASK00	数値変換用変数(LSTAG00から数値)
	TASK01	数値変換用変数(LSTAG00から数値)
...		
LTAG10	TASK00	LSTAG00(変換用変数) 文字列長
	TASK01	LSTAG05(変換用変数) 文字列長
...		

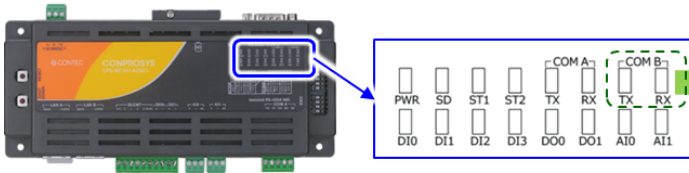
STAG番号	TASK No.	内容
STAG00	TASK00	入力文字列(HMIにて入力)
STAG01	TASK00	送信用Binary値
...		
STAG15	TASK00	変換確認用文字列
...		
STAG20	TASK01	表示用変数(パターン1)
STAG25	TASK01	表示用変数(パターン2)
STAG30	TASK01	表示用変数(パターン3)
...		

LSTAG番号	TASK No.	内容
LSTAG00	TASK00	変換用変数(入力文字列から1Byte分抜粋)
	TASK01	受信電文(Link0からの受信)
...		
LSTAG05	TASK00	Binary文字作成用変数(LTAG05からBinary値)
	TASK01	ASCII文字列作成用変数(LTAG05からASCII文字列)
...		
LSTAG20	TASK01	表示用変数(パターン1計算用)
LSTAG25	TASK01	表示用変数(パターン2計算用)
LSTAG30	TASK01	表示用変数(パターン3計算用)
...		

7. 参考情報

◇通信出来ない場合は？

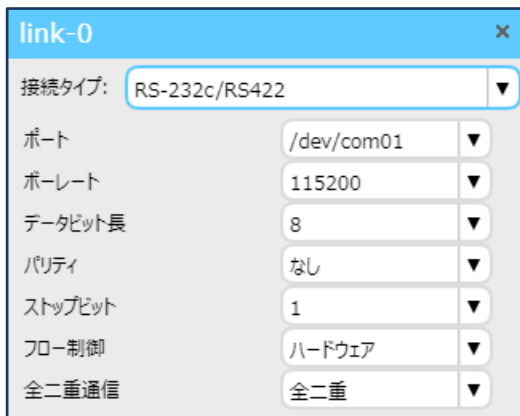
- ・送信受信LEDは点灯しているでしょうか？



Txが点灯していない場合
⇒送信されていません。COMポート指定が正しいか？
タスクが動いているか？等を確認下さい。

Rxが点灯していない場合
⇒受信ができていない状態です。
COMポートが正しいか、結線が正しいか等をご確認下さい。

- ・シリアル通信設定は合っているでしょうか？
⇒パリティ・ボーレート・データビット等を確認下さい。



- ・ご利用のシリアル通信ソフトウェアはバイナリ通信に対応しているでしょうか？
ソフトウェアによっては、『0x00』以降は、送信・受信できないソフトもございますのでご注意下さい。
- ・電文途中で受信する、意図せぬ箇所から変換してしまうといった場合には、『無通信タイムアップ』、『固定長受信』、『デリミタ指定』等の設定を活用下さい。

株式会社コンテック 〒555-0025 大阪市西淀川区姫里3-9-31

<https://www.contec.com/>

本製品および本書は著作権法によって保護されていますので無断で複写、複製、転載、改変することは禁じられています。

2018年12月07日改訂