

表 題 QCPU (Q モード) PID 制御命令サンプル説明書

概 要

QCPU搭載の不完全微分によるPID制御命令 (S.PIDCONT) を使用してPID制御を行うための、サンプルラダープログラムとサンプル画面を説明します。

GOT1000



MELSEC-QCPU



フェースプレートやチューニング画面を備えた  
すぐに使える「GOT1000画面サンプル」

GOT1000サンプル画面に対応した  
「PID制御命令サンプルプログラム」

サンプル画面:S.PIDCONT\_SVGA.GTE

サンプルプログラム:S.PIDCONT\_PRG

目 次

1. サンプルの概要.....	4
2. 使用手順.....	5
3. PID制御命令サンプルプログラムについて.....	6
4. PID制御命令サンプル画面について.....	12
付ー 1.使用デバイス範囲の変更方法.....	15
付ー 2.使用ループの追加方法.....	16

作成	2009 年 01 月 19 日
----	------------------

改定履歴

日付	改 定 内 容
2009 年 1 月	初版作成。

## ● ご利用条件とご注意 ●

(ご使用前に必ずお読みください)

本サンプルは、下記利用条件と注意事項をご了承の上、ご利用ください。

1. 本説明書で紹介するプログラム、画面はサンプルです。お客様のシステムで問題なく動作するか十分確認してからご使用ください。
2. 本サンプルを利用することによって生じた如何なる損害も当社は補償をいたしません。
3. 本サンプルに関して技術的サポートは行っておりませんので、ご承知の上、ご使用をお願いいたします。
4. 本サンプルは、予告なしに削除、内容の変更を行うことがあります。

## ● 安全上のご注意 ●

(ご使用前に必ずお読みください)

本製品のご使用に際しては、本説明書および本説明書で紹介している関連マニュアルをよくお読みいただくと共に、安全に対して十分に注意を払って、正しい取扱いをしていただくようお願いいたします。

本説明書で示す注意事項は、本製品に関するもののみにについて記載したものです。シーケンサシステムとしての安全上のご注意に関しては、CPUユニットのユーザーズマニュアルを参照してください。


この●安全上のご注意●では、安全注意事項のランクを「危険」、「注意」として区分してあります。

### 危険

取扱いを誤った場合に、危険な状況が起こりえて、死亡または重傷を受ける可能性が想定される場合。

### 注意

取扱いを誤った場合に、危険な状況が起こりえて、中程度の傷害や軽傷を受ける可能性が想定される場合および物的損害だけの発生が想定される場合。

なお、 注意に記載した事項でも、状況によっては重大な結果に結びつく可能性があります。

いずれも重要な内容を記載していますので必ず守ってください。

本マニュアルは必要なときに読めるよう大切に保管すると共に、必ず最終ユーザまでお届けいただくようお願いいたします。

### 【立上げ・保守時の注意事項】

#### 注意

- オンライン操作では、マニュアルを熟読し、十分に安全を確認してから行ってください。  
操作ミスによる機械の破損や事故の原因になります。

## 1. サンプルの概要

本サンプルは、QCPU 搭載の不完全微分による PID 制御命令 (S.PIDCONT) を使用して、PID 制御を行うためのサンプルプログラムとサンプル画面\*1 で構成されています。

本サンプルは、以下の CPU ユニット、GOT\*2 で使用できます。

### 適合機種一覧

機種	形名
シーケンサ	ベーシックモデル QCPU
	Q00JCPU, Q00CPU, Q01CPU (シリアル No. の上 5 桁が “04122” 以降)
	ハイパフォーマンズ QCPU
	Q02CPU, Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU (シリアル No. の上 5 桁が “05032” 以降)
二重化 CPU	Q12PRHCPU, Q25PRHCPU
	ユニバーサルモデル QCPU
	Q00UJCPU, Q00UCPU, Q01UCPU, Q02UCPU, Q03UDCPU, Q04UDHCPU, Q06UDHCPU, Q10UDHCPU, Q13UDHCPU, Q20UDHCPU, Q26UDHCPU, Q03UDECPU, Q04UDEHCPU, Q06UDEHCPU, Q10UDEHCPU, Q13UDEHCPU, Q20UDEHCPU, Q26UDEHCPU
表示器	GOT
	GT16**-S, GT15**-S, SoftGOT1000 (Version 2.93X 以降)

\*1: サンプル画面はサンプルプログラムと対になっていますが、使用デバイス範囲を変更することにより既存のラダープログラムに適用することができます。変更方法は「付-1. 使用デバイス範囲の変更方法」を参照してください。

\*2: サンプルプログラムの PC タイプは Q12H、サンプル画面の GOT タイプは GT15\*\*-S となっています。ご使用の際は、実際の機種に合わせて変更してください。

また、サンプル使用の際は以下の項目に注意してください。

- ・本サンプルは、不完全微分命令 (S.PIDCONT) のみに対応しています。
- ・サンプル画面サイズは 800 x 600 (SVGA) です。
- ・サンプル画面は、オブジェクトスクリプト、ロギング機能を使用しています。そのため、オブジェクトスクリプト、ロギング機能があらかじめ GOT にインストールされている必要があります。
- ・サンプル画面において、GOT のシーケンサとの接続は RS-232C 経由の自局接続となっています。必要に応じて設定を変更してください。
- ・サンプルプログラム、サンプル画面の作成バージョンは以下のようになっています。編集する際は、以降のバージョンをご使用下さい。

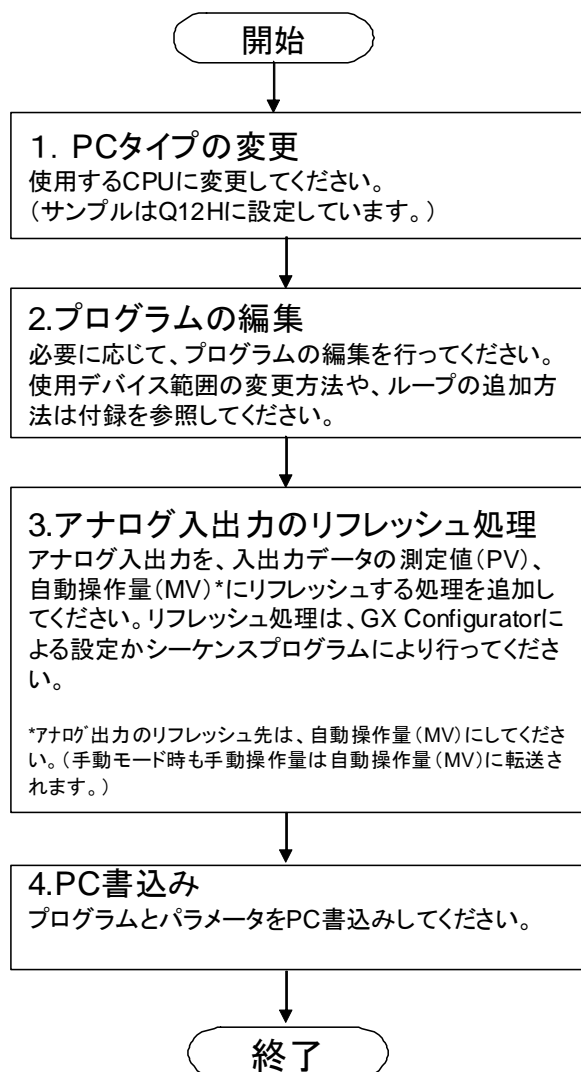
サンプル	作成バージョン
サンプルプログラム	GX Developer Version 8.76E
サンプル画面	GT Designer 2 Version2.93X

本説明書はサンプルを利用する際に最低限必要なことのみ記載しております。関連する詳細情報につきましては、以下のマニュアルを参照ください。

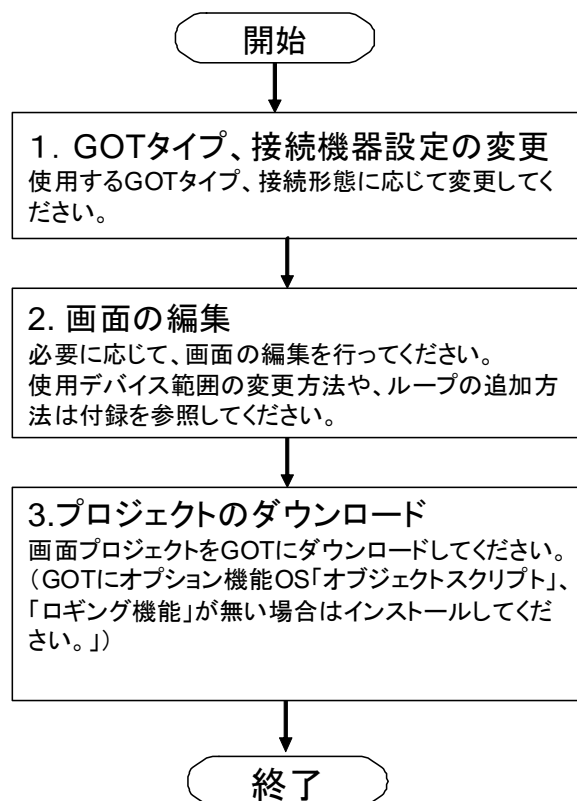
マニュアル名称	マニュアル番号
QCPU (Q モード) /QnACPU プログラミングマニュアル (PID 制御命令編)	SH(名)-080022
GX Developer Version8 オペレーティングマニュアル	SH(名)-080356
GT Designer2 Version2 基本操作・データ転送マニュアル	SH(名)-080508

## 2. 使用手順

### ・サンプルプログラムの使用手順



### ・サンプル画面の使用手順



3. PID制御命令サンプルプログラムについて

(1) 使用データ

PID 制御命令用のデータは以下の 2 種類です。サンプルプログラムの割付先頭デバイスと使用点数、および各データの構造を示します。

・サンプルプログラムの使用データ

データ	説明	割付先頭デバイス <sup>*1</sup>	使用点数 <sup>*2</sup>
PID 制御用データ	PID 制御のパラメータなどの設定を行うためのデータ。	D100	16 点(ワード)
入出力データ	設定値や測定値など PID 演算を行うために設定する入力データおよび演算結果などの出力データ。	D600	33 点(ワード)

<sup>\*1</sup>: 割付デバイスの変更を行う場合は、「付・1. 使用デバイス範囲の変更方法」を参考にしてください。

<sup>\*2</sup>: サンプルプログラムは、No.1 ループ用まで使用しています。ループを追加する場合は、「付・2. 使用ループの追加方法」を参考にしてください。

・各データの構造

①PID制御用データ

(a) 不完全微分の場合

指定デバイス番号+0	使用ループ数	全ループ共通
+1	1スキャンの実行ループ数	
+2	演算式選択	
+3	サンプリング周期(Ts)	
+4	比例定数(Kp)	
+5	積分定数(Ti)	
+6	微分定数(Td)	
+7	フィルタ係数(α)	
+8	操作量下限値(MVLL)	
+9	操作量上限値(MVHL)	
+10	操作量変化率リミット値(ΔMVL)	
+11	測定値変化率リミット値(ΔPVL)	
+12	0 *	
+13	微分ゲイン(Kd)	
+14	0 *	
+15	0 *	
+16	演算式選択	No. 1ループ用 (14ワード)
+17	サンプリング周期(Ts)	
+18	比例定数(Kp)	
+19	積分定数(Ti)	
+20	微分定数(Td)	
+21	フィルタ係数(α)	
+22	操作量下限値(MVLL)	
+23	操作量上限値(MVHL)	
+24	操作量変化率リミット値(ΔMVL)	
+25	測定値変化率リミット値(ΔPVL)	
+26	0 *	
+27	微分ゲイン(Kd)	
+28	0 *	
+29	0 *	
+ (m+0)	演算式選択	No. 2ループ用 (14ワード)
+ (m+1)	サンプリング周期(Ts)	
+ (m+2)	比例定数(Kp)	
+ (m+3)	積分定数(Ti)	
+ (m+4)	微分定数(Td)	
+ (m+5)	フィルタ係数(α)	
+ (m+6)	操作量下限値(MVLL)	
+ (m+7)	操作量上限値(MVHL)	
+ (m+8)	操作量変化率リミット値(ΔMVL)	
+ (m+9)	測定値変化率リミット値(ΔPVL)	
+ (m+10)	0 *	
+ (m+11)	微分ゲイン(Kd)	
+ (m+12)	0 *	
+ (m+13)	0 *	

$m = (n-1) \times 14 + 2$

②入出力用データ

(a) 不完全微分の場合

デバイス番号+0	初回処理フラグ	ライト	No. 1ループ用 入出力データエリア (23ワード)
+1	PID制御用ワークエリア (ユーザ使用不可)	リード/ライト禁止	
+9	設定値(SV)	ライト	
+10	測定値(PV)	リード	
+11	自動操作量(MV)	ライト	
+12	フィルタ後の測定値(PVf)	リード	
+13	手動操作量(MVMAN)	ライト	
+14	手動/自動選択(MAN/AUTO)	ライト	
+15	アラーム(ALARM)	リード/ライト	
+16	No. 1ループ用ワークエリア (ユーザ使用不可)	リード/ライト禁止	
+17	設定値(SV)	ライト	
+18	測定値(PV)	リード	
+19	自動操作量(MV)	ライト	
+20	フィルタ後の測定値(PVf)	リード	
+21	手動操作量(MVMAN)	ライト	
+22	手動/自動選択(MAN/AUTO)	ライト	
+23	アラーム(ALARM)	リード/ライト	
+24	No. 2ループ用ワークエリア (ユーザ使用不可)	リード/ライト禁止	No. 2ループ用 入出力データエリア (23ワード)
+25	設定値(SV)	ライト	
+26	測定値(PV)	リード	
+27	自動操作量(MV)	ライト	
+28	フィルタ後の測定値(PVf)	リード	
+29	手動操作量(MVMAN)	ライト	
+30	手動/自動選択(MAN/AUTO)	ライト	
+31	アラーム(ALARM)	リード/ライト	
+32	No. 3ループ用ワークエリア (ユーザ使用不可)	リード/ライト禁止	
+33	設定値(SV)	ライト	
+34	測定値(PV)	リード	
+35	自動操作量(MV)	ライト	
+36	フィルタ後の測定値(PVf)	リード	
+37	手動操作量(MVMAN)	ライト	
+38	手動/自動選択(MAN/AUTO)	ライト	
+39	アラーム(ALARM)	リード/ライト	
+40	No. nループ用ワークエリア (ユーザ使用不可)	リード/ライト禁止	No. nループ用 入出力データエリア (23ワード)
+41	設定値(SV)	ライト	
+42	測定値(PV)	リード	
+43	自動操作量(MV)	ライト	
+44	フィルタ後の測定値(PVf)	リード	
+45	手動操作量(MVMAN)	ライト	
+46	手動/自動選択(MAN/AUTO)	ライト	
+47	アラーム(ALARM)	リード/ライト	
+48	No. nループ用ワークエリア (ユーザ使用不可)	リード/ライト禁止	
+49	設定値(SV)	ライト	
+50	測定値(PV)	リード	
+51	自動操作量(MV)	ライト	
+52	フィルタ後の測定値(PVf)	リード	
+53	手動操作量(MVMAN)	ライト	
+54	手動/自動選択(MAN/AUTO)	ライト	
+55	アラーム(ALARM)	リード/ライト	

$m = (n-1) \times 23 + 10$

## (2) サンプルプログラム

サンプルは、No.1 ループ用として作成しています。

### ・サンプルでの設定値

サンプルでは、以下のようにデータを設定しています。必要に応じて、変更してください。

#### ①PID 制御用データ

項目	設定例
使用ループ 数	1
1 スキャンの実行ループ 数	1
演算式	1 (逆動作)
サンプリング周期	100 (10ms)
比例定数	100 (0.01)
積分定数	3000 (100ms)
微分定数	0 (10ms)
フィルタ係数	0 (%)
操作量下限値*	0
操作量上限値*	4000*
操作量変化率リミット値*	4000*
測定値変化率リミット値*	4000*
微分ゲイン	800 (0.01)

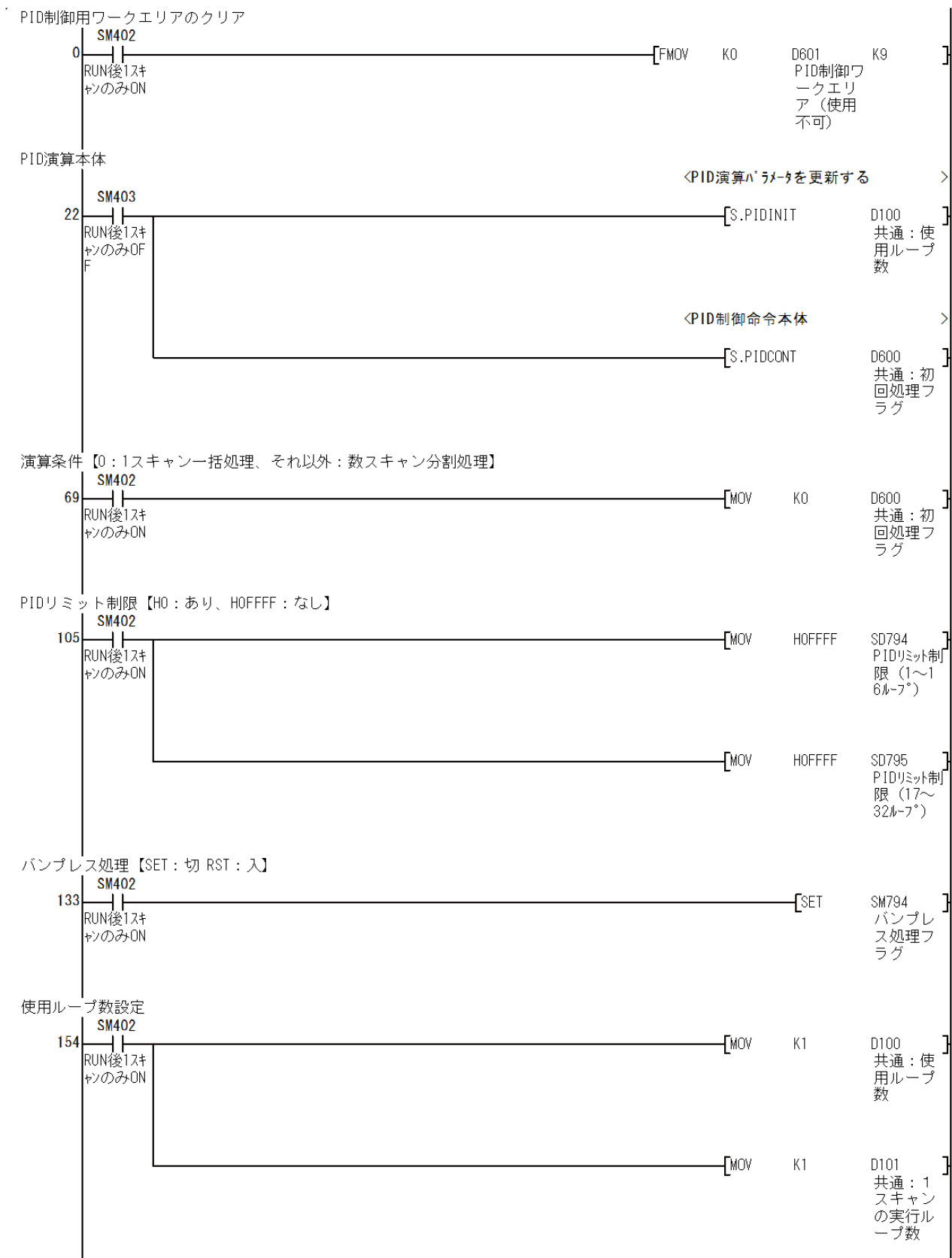
\* : アナログユニットのデジタル値の入出力範囲が 0-4000 であるレンジを基準にサンプルを作成しています。

PV/SV/MV 値の入出力範囲が 0-4000 と異なる場合は必要に応じて変更ください。

#### ②入出力データ

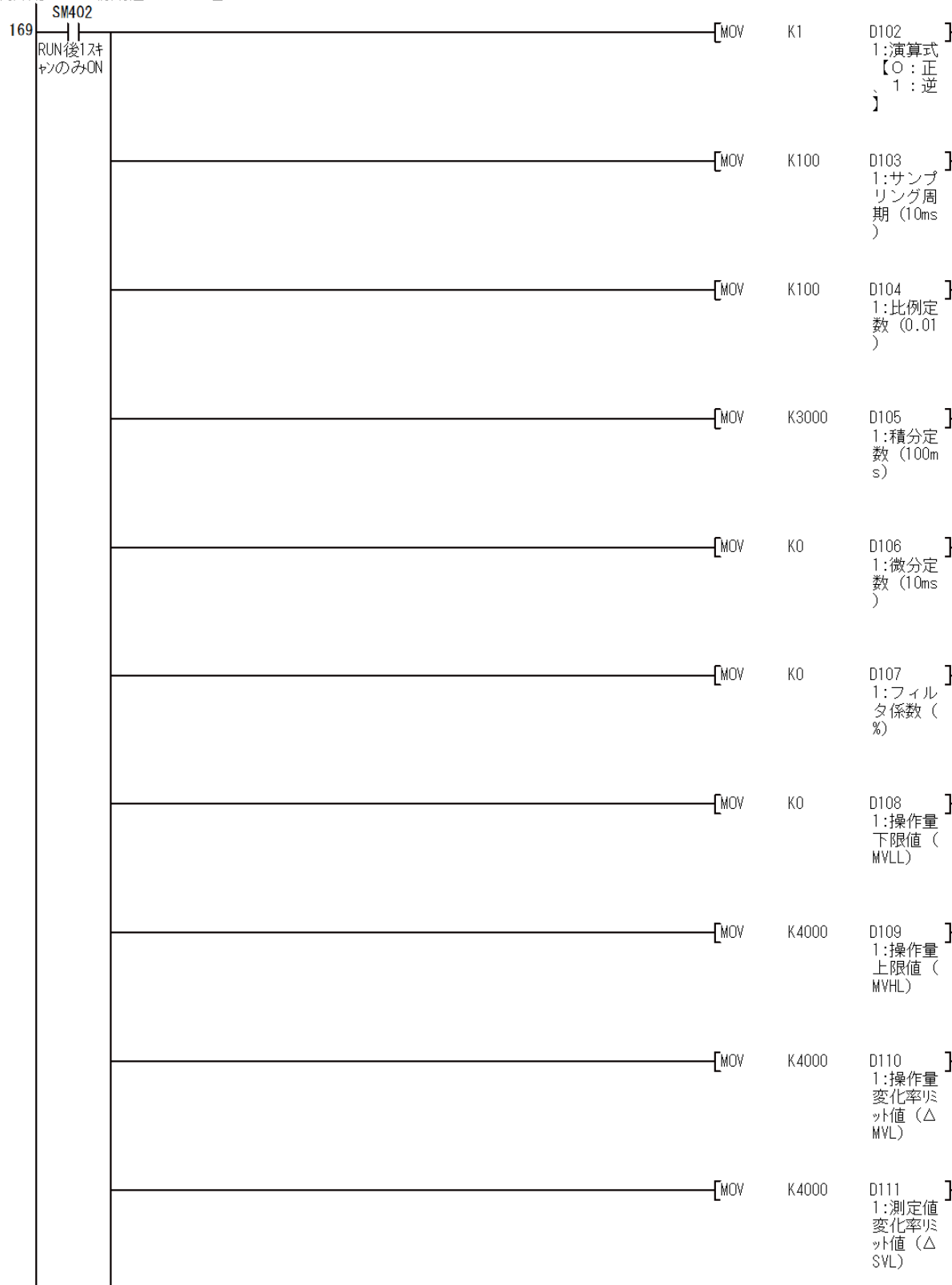
項目	設定例
自動／手動モード	1 (自動)
他の項目	0

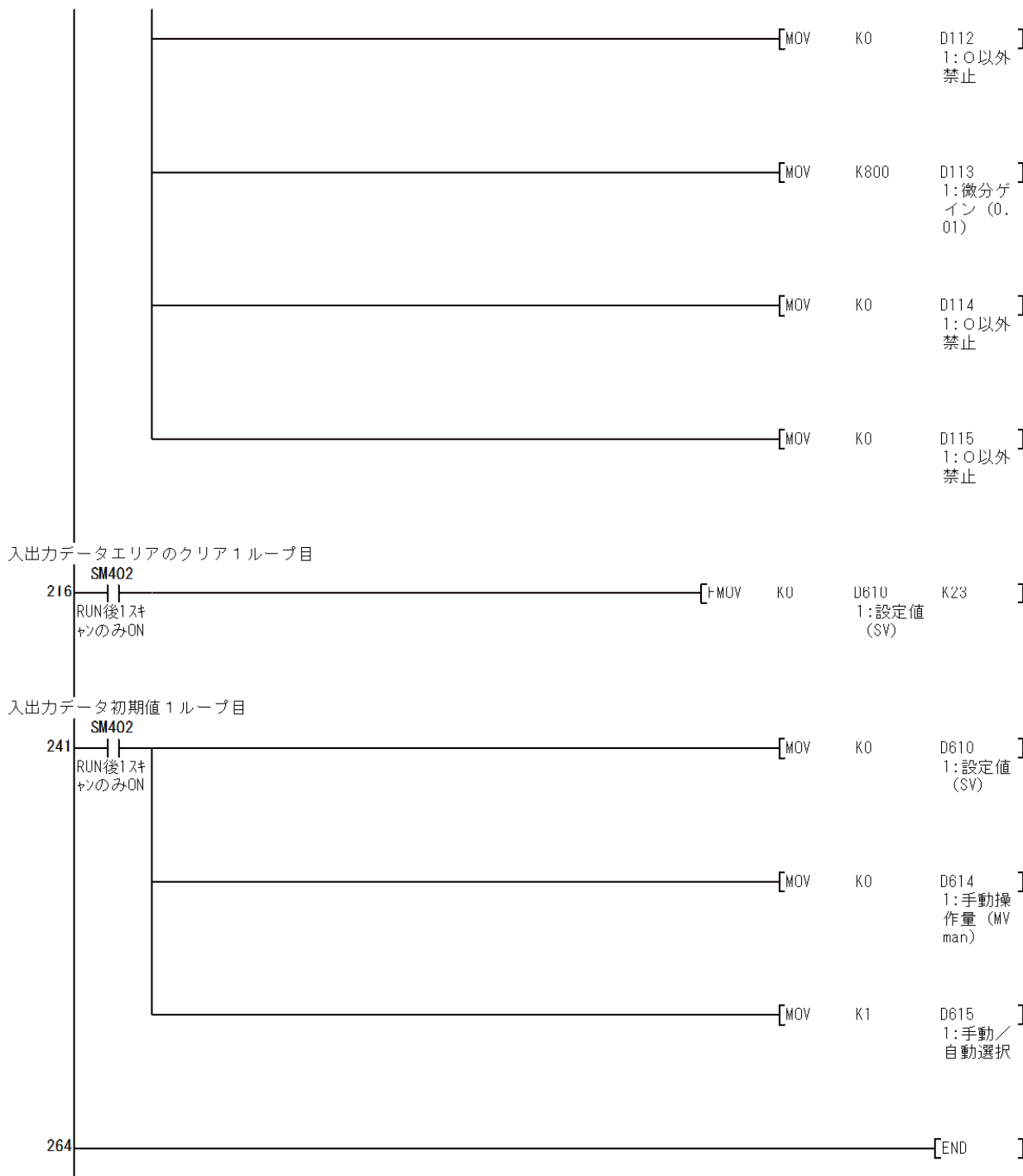
・プログラム





PID制御用データ初期値1 ループ目





## ポイント

### ①アナログ入出力のリフレッシュ処理

アナログ入出力については、入出力データの測定値 (PV)、自動操作量 (MV) \*にリフレッシュする処理を追加してください。リフレッシュ処理は、GX Configurator による設定かシーケンスプログラムにより行ってください。

\*: アナログ出力のリフレッシュ先は、自動操作量 (MV) にしてください。(手動モード時も手動操作量は自動操作量 (MV) に転送されます。)

## ②制御用データ、入出力データの保持と初期化

サンプルでは特殊リレー**SM402** を利用し、**RUN** 後 1 スキャンで制御用データ、入出力データを初期化しています。動作開始後、電源 **OFF** や **CPU** リセットを実施してもこれらの値を保持したい場合は、各データに割り付けたデバイスをラッチするか、割付デバイスをファイルレジスタに変更するなどし、任意のタイミングで初期値を代入するよう変更してください。

4. PID制御命令サンプル画面について

(1) 使用データ

サンプル画面で使用するデータと割付先頭デバイスと使用点数は以下のとおりです。

・サンプル画面の使用デバイス

データ	説明	割付先頭デバイス*1	点数*2
PID 制御用データ	サンプルプログラムと同じです。	D100	16 点(ワード)
入出力データ	サンプルプログラムと同じです。	D600	33 点(ワード)
画面切換えデバイス	ベース画面用切換えデバイスです。	GD30000	1 点(ワード)
グラフ情報格納デバイス	チューニング画面のヒストリカルトレンドグラフで過去のデータを表示する際の格納用として使用されます。	GD30100	10 点(ワード)
インテグレーションスイッチ用トリガデバイス	インテグレーションスイッチ動作を実現するためのデバイスです。	GB30100	8 点(ビット)

\*1：割付デバイスを変更したい場合は、「付-1. 使用デバイス範囲の変更方法」を参考にしてください。

\*2：サンプル画面もサンプルプログラム同様に No.1 ループ用まで使用しています。ループ追加の方法は、「符-2. ループ追加の方法」を参照してください。ただし、画面切換えデバイスは使用ループ数によらず 1 点（ワード）固定です。

サンプル画面の GOT の使用リソースは以下のとおりです。

・サンプル画面の使用リソース

項目	内容	番号/ID
ベース画面	コントロールパネル	30000
	No.1 ループ 用チューニング画面	30001
ウィンドウ画面	No.1 ループ 用フェイスプレート	30001
ロギング設定	No.1 ループ 用ロギング設定	100

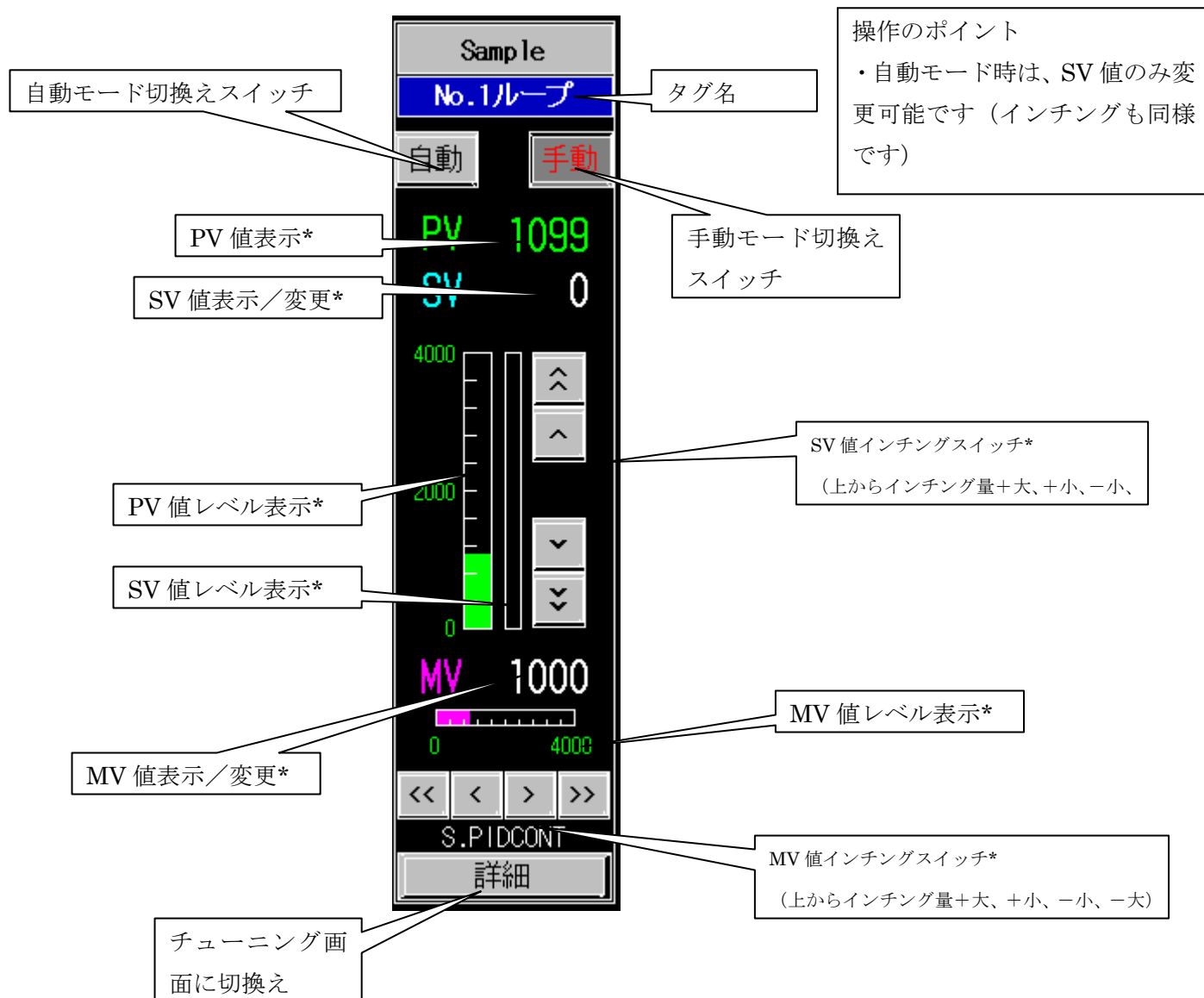
(2) サンプル画面

サンプル画面の遷移と各画面内容を示します。

・遷移

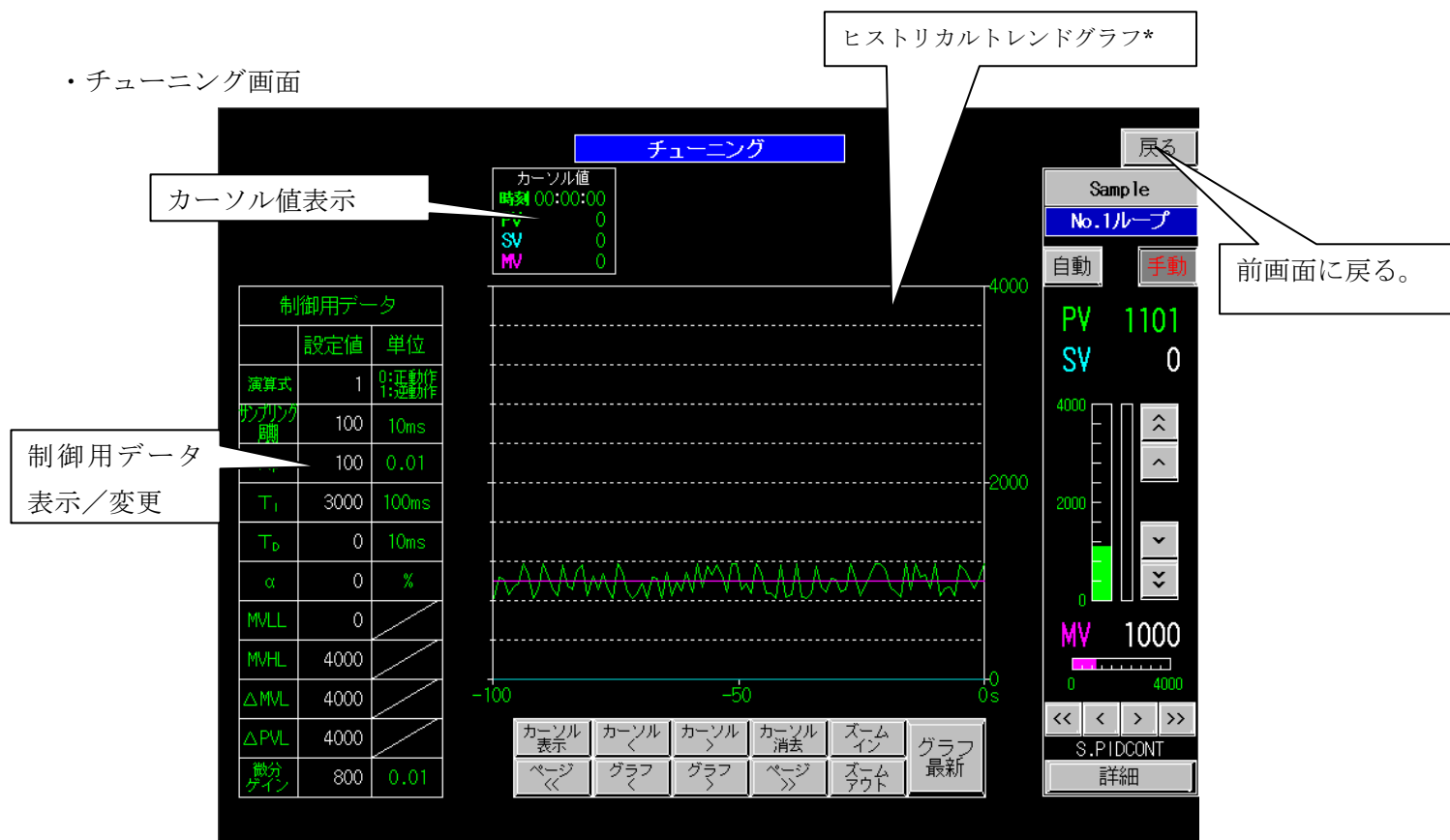


・フェースプレート



\*: アナログユニットのデジタル値の入出量範囲が 0-4000 であるレンジを基準にサンプルを作成しています。PV/SV/MV 値の入出量範囲が 0-4000 と異なる場合は必要に応じて変更ください。

・チューニング画面



\*: アナログユニットのデジタル値の入出量範囲が 0-4000 であるレンジを基準にサンプルを作成しています。PV/SV/MV 値の入出量範囲が 0-4000 と異なる場合は必要に応じて変更ください。

・コントロールパネル



## 付－ 1.使用デバイス範囲の変更方法

サンプルプログラム、サンプル画面の使用デバイスの変更方法を説明します。

### (1) サンプルプログラムの変更方法

サンプル（1 ループ分）を GX Developer のデバイス一括置換機能により、以下のデータをデバイス置換します。

項目*1	旧デバイス (サンプルの割付先頭デバイス)	新デバイス	置換点数 (10 進) *2	点数の計算方法(n:ループ数)
制御用データ	D100	任意のデバイス	16	$2+14\times n$
入出力データ	D600	任意のデバイス	33	$10+23\times n$

\*1：制御用データ、入出力データで使用するデバイスは重複しないようにしてください。

\*2：サンプルは 1 ループ分です。ループを追加した場合は、表の計算式に従い点数を再計算してください。

ループの追加方法は「付－ 2. 使用ループの追加方法」で説明しています。

アナログ入出力のリフレッシュ処理を GX Configurator で設定している場合は、GX Configurator にて置換後先頭デバイスを設定しなおしてください。

### (2) サンプル画面の変更方法

サンプル（1 ループ分）を GT Designer2 のデバイス一括変更機能により、以下のデータ項目をデバイス置換します。なお、GT Designer2 のデバイス一括変更機能は、一度に一括置換できるデバイスの部品が限られているため、対象を「全画面」「共通設定」「スクリプトテキスト（オブジェクトスクリプト）」の 3 回に分けて実行する必要があります。

項目*1	デバイス	変換前 (サンプルの割付先頭デバイス)	変換後 (置換後先頭デバイス)	点数*2	点数の計算方法 (n:ループ数)
制御用データ	ワード	D100	任意のデバイス	16	$2+14\times n$
入出力データ	ワード	D600	任意のデバイス	33	$10+23\times n$
画面切り替えデバイス	ワード	GD30000	任意のデバイス	1	1(固定)
グラフ情報格納デバイス	ワード	GD30100	任意の GD デバイス	10	$10\times n^{*3}$
インテグレーションスイッチ用トリガデバイス	ビット	GB30100	任意の GB デバイス	8	$8\times n^{*3}$

\*1：各データで使用するデバイスは重複しないようにしてください。

\*2：サンプルは 1 ループ分です。ループを追加した場合は、表の計算式に従い点数を再計算してください。

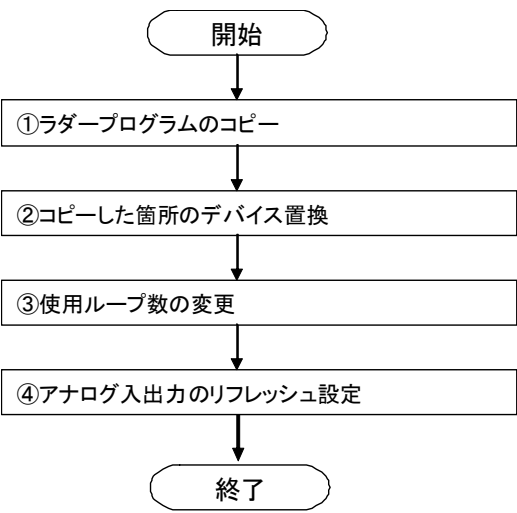
ループの追加方法は「付－ 2. 使用ループの追加方法」で説明しています。

\*3：グラフ情報格納デバイスとインテグレーションスイッチ用トリガデバイス使用するデバイスは、使用ループ数分で連続である必要はありませんが、連続で無い場合は、使用ループ毎に個別にデバイス置換を行ってください。

付－ 2.使用ループの追加方法

使用ループを追加する際のサンプルプログラム、サンプル画面の変更方法を説明します。ここでは、サンプル（No.1 ループ用）に、さらに 1 ループ（No.2 ループ用）を追加する例で説明します。

- （１）サンプルプログラムの変更方法
- サンプルプログラムの変更方法は以下の手順で行います。



①ラダープログラムのコピー

サンプルプログラムの”ループ 1 用データの設定”(169～263 ステップ目) をコピーし、264 ステップ目に貼り付けます。

②コピーした箇所のデバイス置換

貼り付けた部分を選択した状態で、メニューの[検索/置換(S)]→[デバイス置換(R)...]を選択し、デバイス置換を行います。デバイス置換はPID制御用データとPID入出力データにわけて行います。本例は、No.1 ループ用のデータ設定箇所をコピーしてNo.2 ループ用のものとするので、デバイス置換は 以下のようになります。

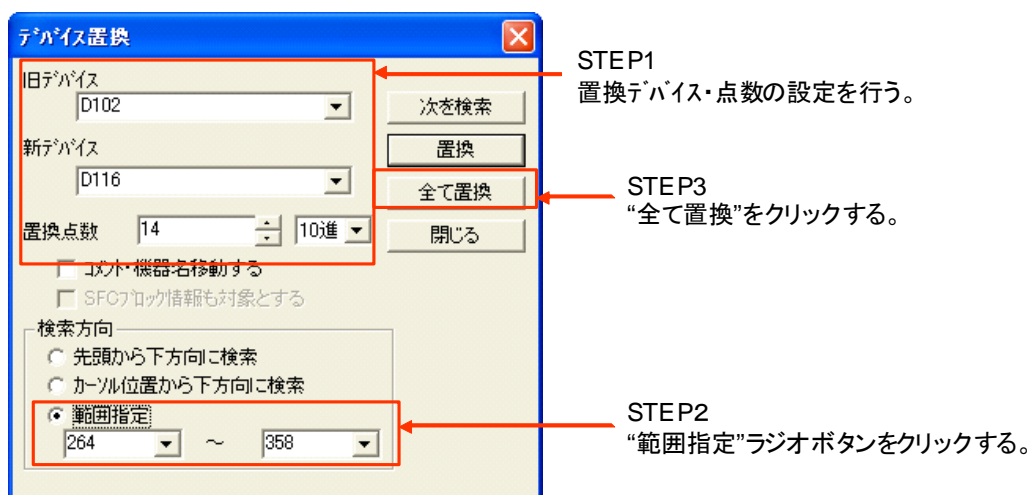
デバイス置換例

項目	旧デバイス (No.1 ループ 先頭デバイス)	新デバイス (No.2 ループ 先頭デバイス) *	置換点数 (10 進)
制御用データ	D102	D116	14 (1 ループ 固定)
入出力データ	D610	D633	23 (1 ループ 固定)

\*制御用データ、入出力データは連続である必要があります。そのため、新デバイス（No.2 ループ用）は、旧デバイス（No.1 ループ用）に置換点数を追加したものになります。データ構造の詳細は 3（1）を参照してください。



・デバイス置換画面入力例



③使用ループ数の変更

165 ステップ目～の回路を以下のように変更します。使用ループ数の詳細は、QCPU(Q モード)/QnACPU プログラミングマニュアル (PID 制御命令編) を参照ください。

MOV K1 D100→MOV K2 D100 (使用ループ数)

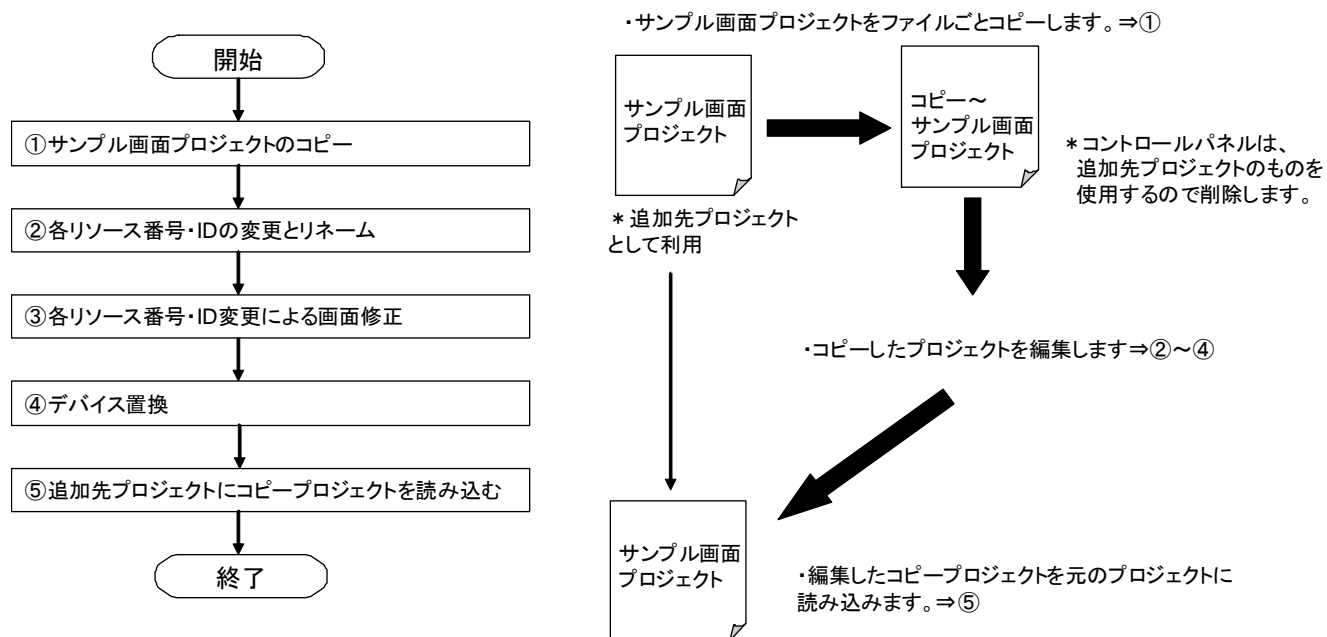
MOV K1 D101→MOV K2 D101 (1 スキャンの使用ループ数)

④アナログ入出力のリフレッシュ処理

追加したループの測定値 (PV)、自動出力値 (MV) 対して、アナログ入出力をリフレッシュする処理を追加してください。リフレッシュ処理は、GX Configurator による設定かシーケンスプログラムにより行ってください。

## (2) サンプル画面の変更方法

サンプル画面の変更方法は以下の手順で行います。



### ① サンプル画面プロジェクトのコピー

サンプル画面プロジェクトをコピーして、コピープロジェクトを GT Designer2 にて開いてください。コントロールパネルは、追加先プロジェクトのものを使用するため、この時点でコピープロジェクトのコントロールパネルは削除してください。

### ② 各種リソース番号・ID の変更とリネーム

リソース毎に実施してください。以下に変更例を示します。

(変更後のリソース番号・ID、リネームは任意のもので設定可能です。)

#### ・ベース画面（チューニング画面）の変更例

項目	変更前	変更後
ベース画面番号	30001	30002
画面名称	No.1 ループ用チューニング画面	No.2 ループ用チューニング画面

#### ・ウィンドウ画面（フェースプレート）の変更例

項目	変更前	変更後
ウィンドウ画面番号	30001	30002
画面名称	No.1 ループ用フェースプレート	No.2 ループ用フェースプレート

#### ・ロギングの変更例

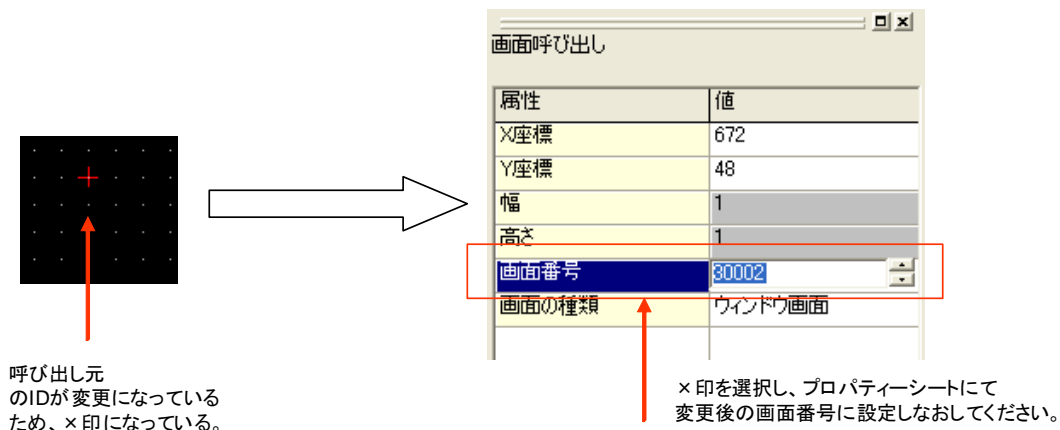
項目	変更前	変更後
ロギング ID	100	101
ロギング名称	No.1 ループ用ロギング設定	No.2 ループ用ロギング設定

### ③各リソース番号・ID 変更による修正

各リソース番号・ID 変更による修正を行います。

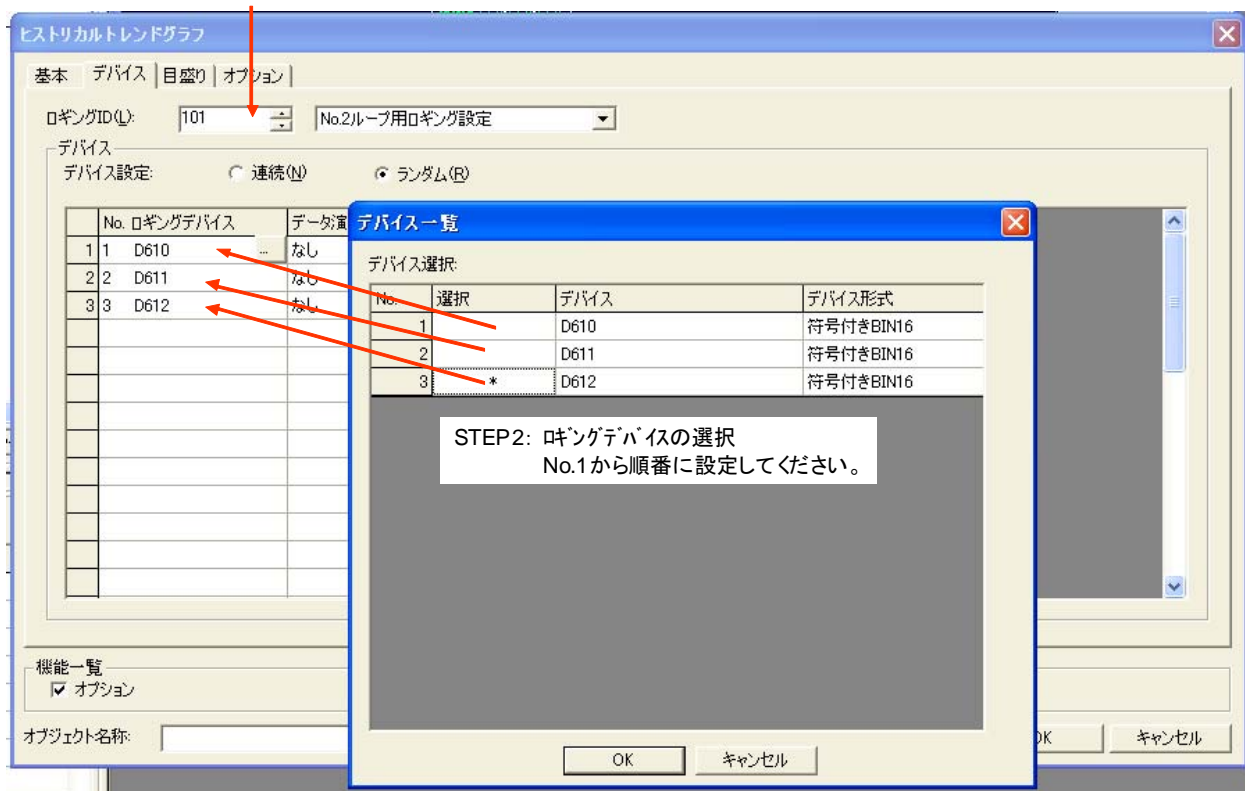
#### ・チューニング画面

画面呼び出し元のウィンドウ番号が変更になったため、フェースプレートに配置している箇所が、下図のような×印になっています。×印を選択し、プロパティシートで画面呼び出しの再設定をしてください。



ヒストリカルトレンドグラフを選択し、以下のようにロギング ID とロギングデバイスを再設定してください。

#### STEP1: ロギングIDの再設定



#### ・フェースプレート

詳細画面スイッチの属性設定にて遷移先を変更後のベース（チューニング）画面の画面番号に再設定してください。

#### ④デバイス置換

デバイス置換を実施します。(実施前にここまでの作業のバックアップを取っておくことをお勧めします。)

本例では以下のように行います。(No.1 ループから No.2 ループへの変更)

項目	デバイス	変換前	変換後	点数
PID 制御用データ	ワード	D102	D116*1	14
入出力データ	ワード	D610	D633*1	23
グラフ情報格納用デバイス	ワード	GD30100	GD30110	10
インテグレーション用デバイス	ビット	GB30100	GB30108	8

\*1: 制御用データ、入出力データは連続である必要があります。そのため、新デバイス (No.2 ループ用) は、旧デバイス (No.1 ループ用) に置換点数を追加したのになります。データ構造の詳細は 3 (1) を参照してください。

メニューより、[ツール(T)]→[一括変更(B)]→[デバイス一括変更(D)...]を選択し、下図のようにデバイス置換を実施してください。

#### ・デバイス置換の変更例

デバイス一括変更

属性  
☒ デバイス(D)    ☐ 色(C)    ☐ 図形(S)    ☐ CH No.(H)

対象  
☒ 全画面(A)    ☐ 編集画面内(E)    ☐ 画面範囲内(N): From: 1 To: 32767    ☐ カテゴリ(C): スイッチ    ☐ 選択範囲内(T)    ☐ 共通設定(M) (画面単位の設定は除く)    ☐ スクリプトテキスト(P): 全スクリプト

検索(E)    変更(R)    削除    クリア(L)

	デバイス	変換前	変換後	点数
1	ワード	D102	D116	14
2	ワード	D610	D633	23
3	ワード	GD30100	GD30110	10
4	ビット	GB30100	GB30108	8
5	ビット			1

閉じる(O)

#### STEP3

“変更”をクリックする。  
クリック後は、STEP2に戻り、次の対象を選択する。

#### STEP2

対象を選択する。

- ・全画面
- ・共通設定
- ・スクリプトテキスト (オブジェクトスクリプト)

#### STEP1

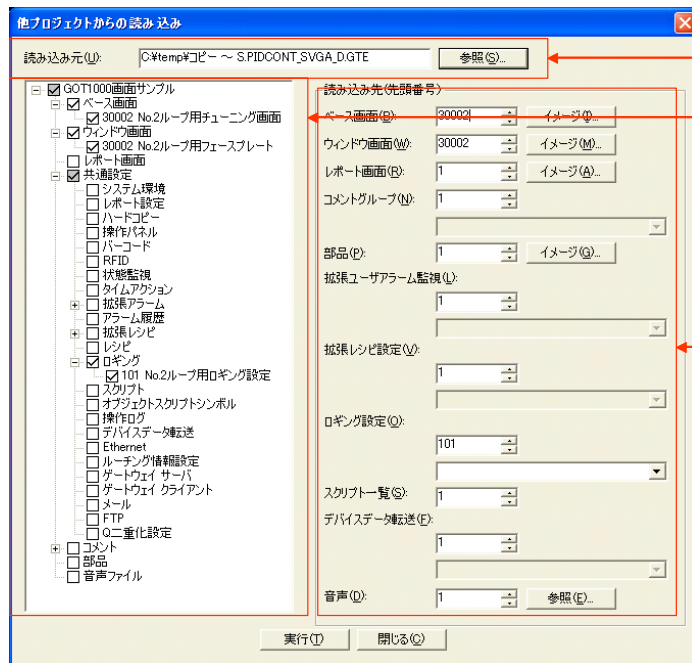
置換デバイスの設定を行う。

# ⑤追加先プロジェクトに読み込む

追加先プロジェクトを開いて、追加ループ分のリソースを読み込みます。

メニューより、[プロジェクト(P)]→[他プロジェクト読み込み(I)...]を選択してください。

以下のように読み込みます。



STEP1: 手順①で作成したプロジェクトを選択します。

STEP2: 読み込むリソースを選択します。

- ・ベース画面(チューニング画面)
- ・ウィンドウ画面(フェースプレート)
- ・ロギング設定

STEP3: 読み込むリソースの先頭番号を指定します。

手順②で変更した番号にしてください。

- ・ベース画面(チューニング画面)
- ・ウィンドウ画面(フェースプレート)
- ・ロギング設定

\*読み込みと重複しないようにしてください。

後は、コントロールパネルに、追加したループのフェースプレートを画面呼び出しにより追加することで完成です。