≪ 1. 概要 ≫

本書は、アナログデータを任意の記録周期毎に CSV ファイル形式で記録･保存する、750-841（WAGO-IO-Pro CAA）用サンプルプログラムについて説明したドキュメントです。後半部ではプログラムの詳細解説と併せて、750-841 内蔵のリアルタイムクロック（RTC）を利用する方法、フラッシュメモリにファイルを作成･更新･削除する手順を簡単に説明します。

≪ 2. プログラムの保存とライブラリの確認 ≫

① プログラムはダウンロード後に解凍（自己解凍形式）し、下記保存先フォルダへコピーして下さい。

DataLogCSV_4ch.pro →（保存先）C:¥Program Files¥WAGO Software¥CoDeSys V2.3¥Projects¥

② ライブラリはデフォルトで読み込まれるライブラリの他に、次のライブラリ（2 種）を使用しています。

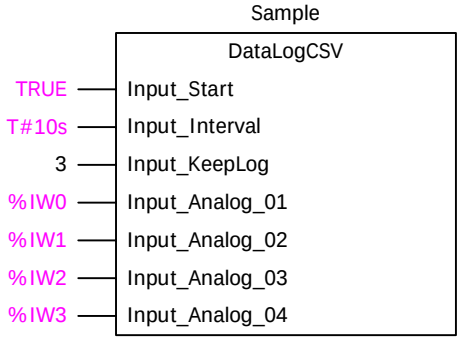
- SysLibRtc.lib ……　リアルタイムクロックに関するライブラリ
- SysLibFile.lib ……　ファイルの作成･更新･削除に関するライブラリ

≪ 3. プログラム仕様 ≫

アナログ入力データを定周期毎に測定し、フラッシュメモリ内の CSV ファイルに記録･保存します。記録周期、保存期間はある範囲内で任意に設定可能で、保存期間を過ぎた古いデータは自動的に削除します。（詳細は下記参照）

- アナログ入力データは 4 点まで指定可能、実入力以外でも Word 型データであれば指定可能
- 1 日分のデータは日付名のファイル 1 個に記録され、日付が変わった際に新しいファイルを作成
- 記録周期の設定：10 秒以上の任意の値に設定可能
- 保存期間の設定：1 ～ 10 日の間で設定可能
- CSV ファイルは “Root ¥ PLC ¥” に保存（デフォルト設定時 … User：admin, Pass：wago）

≪ 4. 入力パラメータの説明 ≫



入力パラメータ	型	説　明
Input_Start	BOOL	TRUE 時にのみ FB の動作が有効 （FALSE 時は記録動作を休止）
Input_Interval	TIME	記録周期 10秒（T#10s）以上に設定
Input_KeepLog	INT	保存期間 1 ～ 10 日間、整数値で設定
Input_Analog_01 - 04	WORD	記録･保存対象のデータを指定

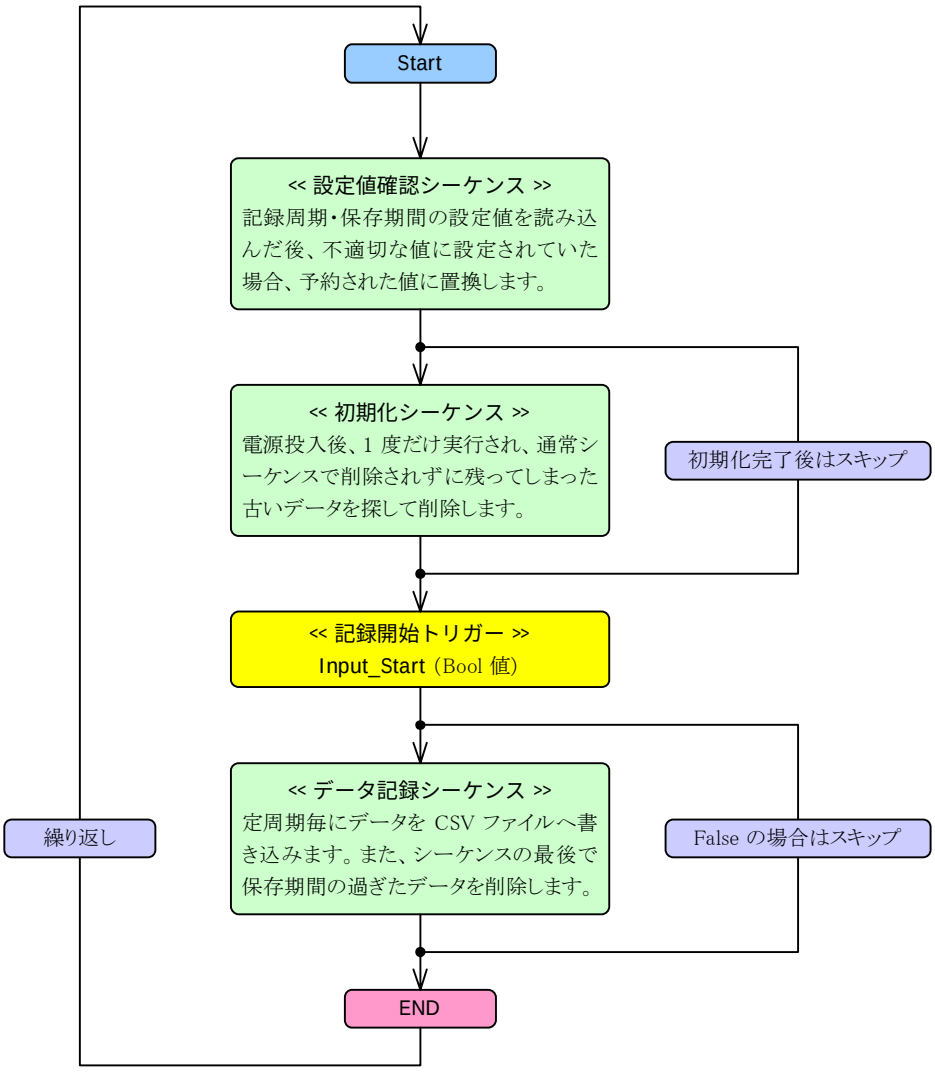
≪ 5. CSV ファイルのフォーマット ≫

カンマ区切りの CSV ファイルに記録します。1 レコード（記録周期 1 回分）につき 1 行で、新しいデータは最終行に追加されます。（項目別は下記参照）

・A ～ B 行：タイムスタンプ
　A 行が日付、B 行が時刻
・C ～ F 行：アナログ入力データ（測定値）
　左から 1ch･2ch･3ch･4ch の順で 4ch 分

	A	B	C	D	E	F
1	2006/8/7	18:37:12	65535	65535	65535	65535
2	2006/8/7	18:37:23	65535	65535	65535	65535
3	2006/8/7	18:37:33	65535	65535	65535	65535
4	2006/8/7	18:37:43	65535	65535	65535	65535
5	2006/8/7	18:37:53	65535	65535	65535	65535
6	2006/8/7	18:38:03	65535	65535	65535	65535
7	2006/8/7	18:38:13	65535	65535	65535	65535
8	2006/8/7	18:38:23	65535	65535	65535	65535
9	2006/8/7	18:38:33	65535	65535	65535	65535
10	2006/8/7	18:38:43	65535	65535	65535	65535
11	2006/8/7	18:38:53	65535	65535	65535	65535
12	2006/8/7	18:39:03	65535	65535	65535	65535
13	2006/8/7	18:39:13	65535	65535	65535	65535
14	2006/8/7	18:39:23	65535	65535	65535	65535
15	2006/8/7	18:39:33	65535	65535	65535	65535

≪ 6. プログラムのフロー概略図 ≫



≪ 7. プログラムの詳細解説 ≫

```
0001  (* Data set sequence *)
0002
0003  IF Input_Interval < T#10s THEN
0004      Interval := T#10s;
0005  ELSE
0006      Interval := Input_Interval;
0007  END_IF
0008
0009  IF Input_KeepLog < 1 THEN
0010      KeepLog := 1;
0011  ELSE
0012      IF Input_KeepLog > 10 THEN
0013          KeepLog := 10;
0014      ELSE
0015          KeepLog := Input_KeepLog;
0016      END_IF
0017  END_IF
0018
0019  (************************************************************)
0020  (* Initialization sequence *)
0021
0022  IF Flag_01 = FALSE THEN
0023
0024      IF Flag_02 = FALSE THEN
0025
0026          n := KeepLog + 1;
0027
0028          RealTimeClock := SysRtcGetTime (Dummy := TRUE);
0029              RtcTemp_02 := DT_TO_DATE (RealTimeClock);
0030              Today := DATE_TO_DWORD (RtcTemp_02);
0031
0032          Flag_02 := TRUE;
0033
0034      ELSE
0035
0036          IF n < KeepLog + 365 THEN
0037
0038              Temp_01 := Today - (86400 * n);
0039              Temp_02 := DWORD_TO_DATA (Temp_01);
0040              Temp_03 := DATE_TO_STRING (Temp_02);
0041              Temp_04 := MID (Temp_03, 10, 3);
0042
```

0003 ～ 0007 行
記録周期の設定値が 10 秒以下に設定された場合、10 秒（T#10s）を強制的にアサインします。

0009 ～ 0017 行
保存期間の設定値が 1 日以下に設定された場合は 1 日を、10 日以上に設定された場合は 10 日を強制的にアサインします。

【F】SysRtcGetTime
750-841 内蔵のリアルタイムクロックの時刻をプログラム中で読み込むためのファンクションです。Dummy は BOOL 値で、TRUE 時にファンクションが有効になります。得られる時刻は DT 型のデータです。
（DT#YYYY-MM-DD-hh:mm:ss）

0029 ～ 0030 行
DT 型で得た時刻を、演算に使用するために DWORD 型に変換しています。DT，DATE 型のデータは『1970 年 1 月 1 日から何秒後』という形で DWORD 型のデータに変換することができます。

0036 ～ 0041 行
削除すべきファイルの日付を演算、サンプルでは保存期間経過後 1 年までのファイルを削除します。ファイル名を指定するために、結果を STRING 型に変換しています。

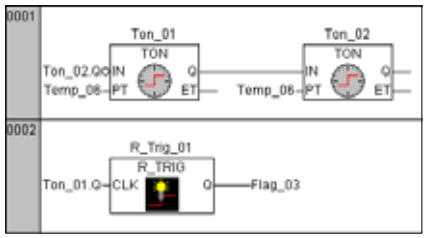
【F】MID
STRING 型データの中から任意の文字列を抜き出すためのファンクションです。対象データ，桁数，位置の順にパラメータを設定します。

```
0043            DelFileName_01 := CONCAT (Temp_04, '.csv');
0044            SysFileDelete (DelFileName_01);
0045
0046            DelFileName_02 := CONCAT (Temp_04, '.CSV');
0047            SysFileDelete (DelFileName_02);
0048
0049            n := n + 1;
0050
0051        ELSE
0052
0053            Flag_01 := TRUE;
0054
0055        END_IF
0056    END_IF
0057 ELSE
0058
0059 (*************************************************************)
0060 (* Clock generate sequence *)
0061
0062    IF Input_Start = TRUE THEN
0063
0064        Temp_05 := Interval / 2;
0065
0066        IF Ton_02.Q = FALSE THEN
0067            Temp_06 := TRUE;
0068        ELSE
0069            Temp_06 := FALSE;
0070        END_IF
0071
0072        Ton_01 (IN := Temp_06, PT := Temp_05);
0073        Ton_02 (IN := Ton_01.Q, PT := Temp_05);
0074
0075        R_Trig_01 (CLK := Ton_01.Q);
0076        Flag_03 := R_Trig_01.Q;
0077
0078 (*************************************************************)
0079 (* Make writebuffer sequence *)
0080
0081        IF Flag_03 = TRUE THEN
0082
0083            RealTimeClock := SysRtcGetTime (Dummy := TRUE);
0084                RtcTemp_01 := DT_TO_STRING (RealTimeClock);
0085                RtcTemp_02 := DT_TO_DATE (RealTimeClock);
```

0043 ～ 0047 行
削除すべきファイル名を指定して、該当ファイルがあれば削除します。サンプルプログラム内では、小文字の拡張子‘.csv’を使用しますが、大文字の拡張子でも正常に削除できるよう、2 パターンのファイル名を生成しています。

【F】SysFileDelete
フラッシュメモリ内のファイルを削除するためのファンクションです。削除したいファイルの名前をパラメータに記述します。

0064 ～ 0076 行
TON と R_TRIG のファンクションを組み合わせて、定周期のクロック発振回路を作成しています。このクロックに基きデータが記録されます。
<< 参考 >>
下図は FBD で記述した例です。



0083 ～ (次頁) 0088 行
前頁と同様にリアルタイムクロックより時刻データを得て、ファイル名、タイムスタンプ、などに利用するためデータ形式を変換します。

```
0086            Date_ST := MID (RtcTemp_01, 10, 4);
0087            Time_ST := MID (RtcTemp_01, 8, 15);
0088            Today := DATE_TO_DWORD (RtcTemp_02);
0089
0090        writebuffer_01 := Date_ST;
0091        writebuffer_02 := Time_ST;
0092
0093        writebuffer_03 := WORD_TO_STRING (Input_Analog_01):
0094        writebuffer_04 := WORD_TO_STRING (Input_Analog_02):
0095        writebuffer_05 := WORD_TO_STRING (Input_Analog_03):
0096        writebuffer_06 := WORD_TO_STRING (Input_Analog_04):
0097
0098        writebuffer_98 := ',';
0099        writebuffer_99 := '$n';
0100
0101 (*************************************************************)
0102 (* Write Log-Data sequence *)
0103
0104        WriteFileName := CONCAT (Date_ST, '.csv');
0105
0106        hfile := SysFileOpen (WriteFileName, 'a');
0107
0108        SysFileWrite (hfile, ADR(writebuffer_01), LEN(wirtebuffer_01);
0109        SysFileWrite (hfile, ADR(writebuffer_98), LEN(wirtebuffer_98);
0110
0111        SysFileWrite (hfile, ADR(writebuffer_02), LEN(wirtebuffer_02);
0112        SysFileWrite (hfile, ADR(writebuffer_98), LEN(wirtebuffer_98);
0113
0114        SysFileWrite (hfile, ADR(writebuffer_03), LEN(wirtebuffer_03);
0115        SysFileWrite (hfile, ADR(writebuffer_98), LEN(wirtebuffer_98);
0116
0117        SysFileWrite (hfile, ADR(writebuffer_04), LEN(wirtebuffer_04);
0118        SysFileWrite (hfile, ADR(writebuffer_98), LEN(wirtebuffer_98);
0119
0120        SysFileWrite (hfile, ADR(writebuffer_05), LEN(wirtebuffer_05);
0121        SysFileWrite (hfile, ADR(writebuffer_98), LEN(wirtebuffer_98);
0122
0123        SysFileWrite (hfile, ADR(writebuffer_06), LEN(wirtebuffer_06);
0124        SysFileWrite (hfile, ADR(writebuffer_98), LEN(wirtebuffer_98);
0125        SysFileWrite (hfile, ADR(writebuffer_99), LEN(wirtebuffer_99);
0126
0127        SysFileClose (hfile);
0128
```

0090 ～ 0099 行
ファイルに書き込むためには、一度文字列（STRING 型データ）をバッファに格納する必要があります。Input_Analog_01 ～ 04 のデータは WORD 型なので、STRING 型に変換してからバッファに格納します。

※ $n は改行コードです。

0104 ～ 0128 行
本サンプルプログラムのメインとなる部分です。ファイルを指定してオープン、データを追加書き込みした後にファイルをクローズしています。hFile は DWORD 型です。

【F】SysFileOpen
ファイルをオープンするためのファンクションです。パラメータには、ファイル名、モードを指定します。

- w ： 上書保存 or 新規作成
- r ： 読取専用
- rw ： 読取後上書 or 新規作成
- a ： 追加書込 or 新規作成

【F】SysFileWrite
ファイルを書き込むためのファンクションです。パラメータにはバッファのアドレス、バッファの長さ(バイト数)を指定します。

【F】SysFileClose
ファイルをクローズするファンクションです。ファイルをクローズしないと、書込･読込が不能になります。

```
0129  (*************************************************************)
0130  (* Delete Log-Data sequence *)
0131
0132              k := KeepLog + 1;
0133
0134          Temp_07 := Today - (86400 * k);
0135          Temp_08 := DWORD_TO_DATE (Temp_07);
0136          Temp_09 := DATE_TO_STRING (Temp_08);
0137          Temp_10 := MID (Temp_09, 10, 3);
0138
0139          DelFileName_03 := CONCAT (Temp_10, '.csv');
0140          SysFileDelete (DelFileName_03);
0141
0142          DelFileName_04 := CONCAT (Temp_10, '.csv');
0143          SysFileDelete (DelFileName_04);
0144
0145        END_IF
0146      END_IF
0147  END_IF
```

0134 ～ 0137 行
削除すべきファイルの日付を演算、ファイル名を指定するために、結果を STRING 型に変換しています。初期化シーケンスと異なり、保存期間が過ぎた直後のデータ 1 つのみを削除します。

0139 ～ 0143 行
前述の通り、小文字と大文字、両方の拡張子に対応するために 2 パターンのファイル名を生成します。

以　上