



Welcome

PLCnext Controlを 使用した CSVファイル出力



構成

- 今回使用するアプリケーションバージョン
 - PLCnext Engineer(1046008)...2024.02LTS
 - AXC F 2152(2404267)...HW 02, FW 2024_05LTS

出力例

- 下記のようなデータをPLCnext Control AXC F2152からcsvファイルを出力する

```
MyData.csv - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)
DT#2024-12-02-07:16:37.20,0,0.10
DT#2024-12-02-07:16:41.20,0,0.10
DT#2024-12-02-07:16:45.20,0,0.10
DT#2024-12-02-07:16:49.20,0,0.10
DT#2024-12-02-07:16:53.20,0,0.10
DT#2024-12-02-07:56:44.70,10,0.10
DT#2024-12-02-07:56:48.70,10,0.10
DT#2024-12-02-07:56:52.70,10,0.10
```

① ② ③

①- csv出力した日時

次以降プログラム例における

変数名: strTimeに該当

②- 整数値(PLCnext Engineerより任意で書き換えて出力)

次以降プログラム例における

変数名: iValueに該当

③- 実数値(浮動小数型)

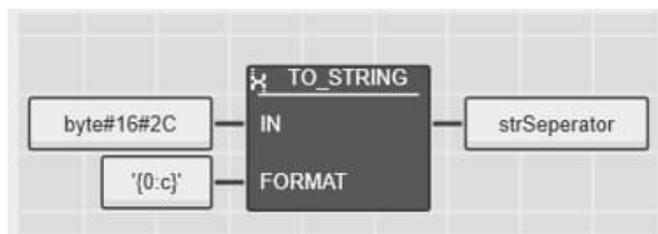
次以降プログラム例における

変数名: LoadCurrent_Ch1_Realに該当

プログラム例

- カンマ記号を定義する

COMPONENTSのSearch画面でTO_STRINGを入力し、TO_STRINGのファンクションブロックをドラック&ドロップしてください



IN– byte#16#2c

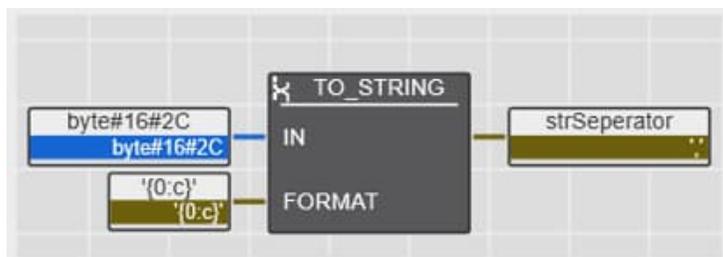
ASCIIコード” , ”にあたる16進数2Cを設定する

FORMAT – {0:c}

INに設定した値が文字もしくは文字列かどうかを設定する

OUT– 文字列型: 変数名(ここではstrSeperator)

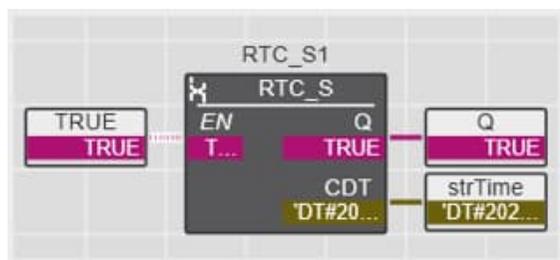
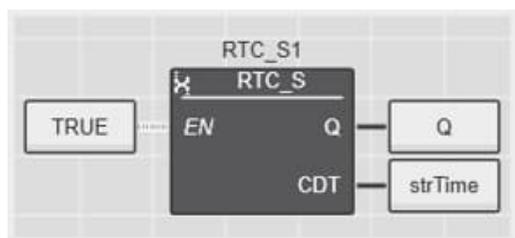
戻り値String型(文字列)、ここでは” , ”(カンマ)が出力



プログラム例

- タイムスタンプ用に時刻を定義する

COMPONENTSのSearch画面でRTC_Sを入力し、RTC_Sのファンクションブロックをドラック&ドロップしてください



EN – bool型: TRUE

TRUEの時、CDTに現在のローカル日時が出力

Q – bool型: 変数名(ここではQ)

CDTに現在日時を出力していることをbool型で出力

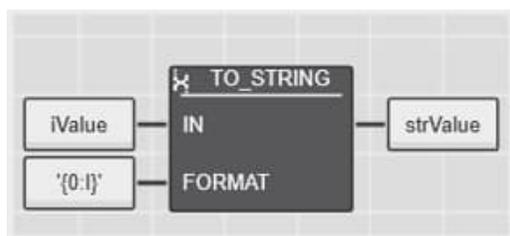
CDT – 文字列型: 変数名(ここではstrTime)

戻り値String型(文字列)、ここでは"DT#現在日時"が出力

プログラム例

- CSVへ出力したい変数を定義する①

COMPONENTSのSearch画面でTO_STRINGを入力し、TO_STRINGのファンクションブロックをドラック&ドロップしてください

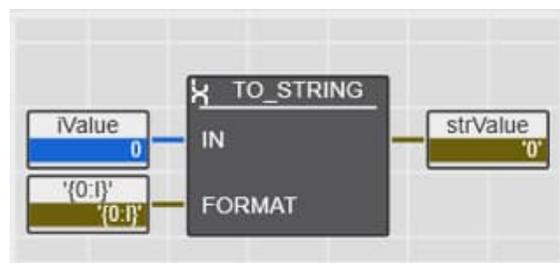


IN- 数値(整数型、小数型、実数型)CSVへ出力したい変数
ここでは整数値(INT型)の値

FORMAT- {0:I}

INに設定している値を整数型を定義

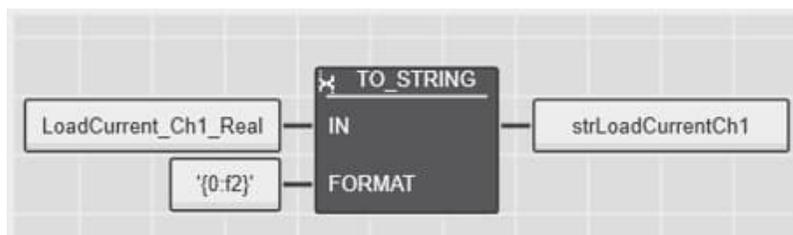
OUT- 文字列型: 変数名(ここではstrLoadCurrentCh1)
戻り値String型(文字列) で整数値を出力



プログラム例

- CSVへ出力したい変数を定義する②

COMPONENTSのSearch画面でTO_STRINGを入力し、TO_STRINGのファンクションブロックをドラック&ドロップしてください



IN- 数値(整数型、小数型、実数型)CSVへ出力したい変数
ここでは実数値(REAL型)の値

FORMAT- {0:f2}

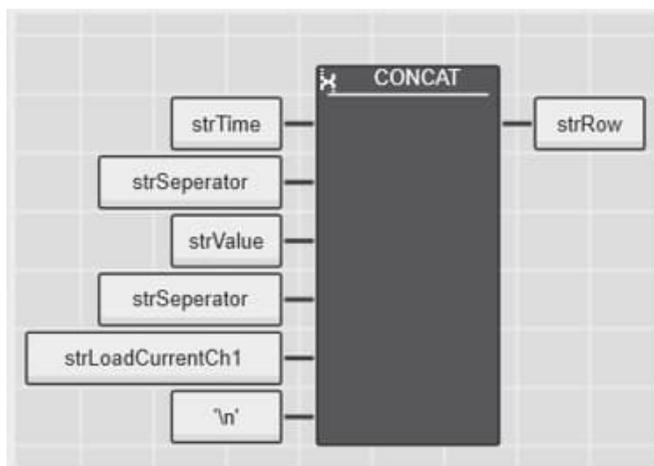
INに設定している値を浮動小数点(小数点下2桁まで)を定義

OUT- 文字列型: 変数名(ここではstrLoadCurrentCh1)
戻り値String型(文字列)、ここでは"0.**"などの
小数点2桁まで文字列で出力

プログラム例

- CSVへ出力したい値を1行データにまとめる定義する

COMPONENTSのSearch画面でCONCATを入力し、CONCATのファンクションブロックをドラック&ドロップしてください



ファンクションブロック左側(IN-)に文字列(String)型を設定する

CSVなので、変数の間にカンマ文字(ここでは変数名strSeperator)を設定する
(", "でも良い)

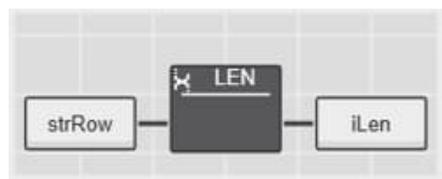
行終わり用に"¥n" (改行)を設定する

OUT- 文字列型: 変数名(ここではstrRow)

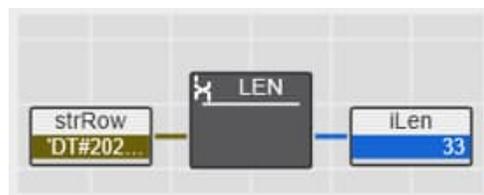
ファンクションブロックの左側に設定した文字列を連結したものを文字列で出力

プログラム例

- CSVへ出力したい1行データの文字列のサイズを出力する
COMPONENTSのSearch画面でLENを入力し、LENのファンクションブロックをドラック&ドロップしてください



IN – 文字列型: 文字サイズを知りたい変数(ここではstrRow)

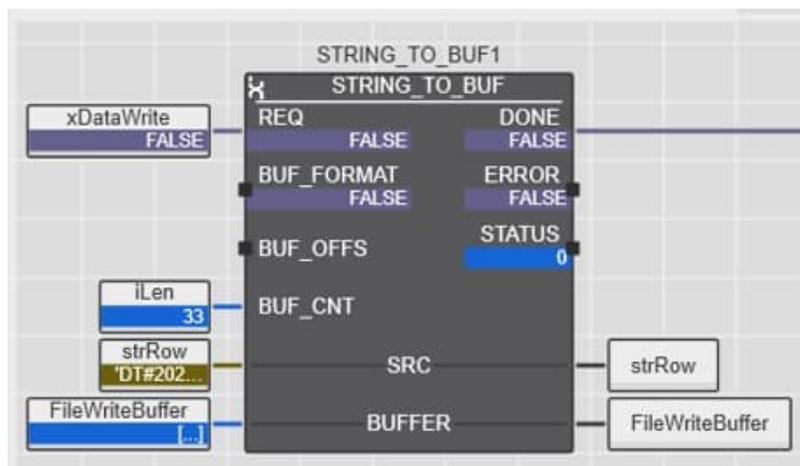


OUT – 整数型: 変数名(ここではiLen)
INで入力された文字列のサイズを出力

プログラム例

- CSVへ出力したい1行データの文字列をバッファする

COMPONENTSのSearch画面でSTRING_TO_BUFを入力し、STRING_TO_BUFのファンクションブロックをドラック&ドロップしてください



REQ- bool型: 変数名(ここではxDataWrite)

TRUEでCSVファイルへ出力したい文字列をバッファへ格納する

BUF_CNT- 整数型: 変数名(ここではiLen)

バッファするサイズ

SRC - 文字列: バッファしたい文字列の変数名(ここではstrRow)

BUFFER- 整数型配列: バッファ格納先変数(ここではFileWriteBuffer)

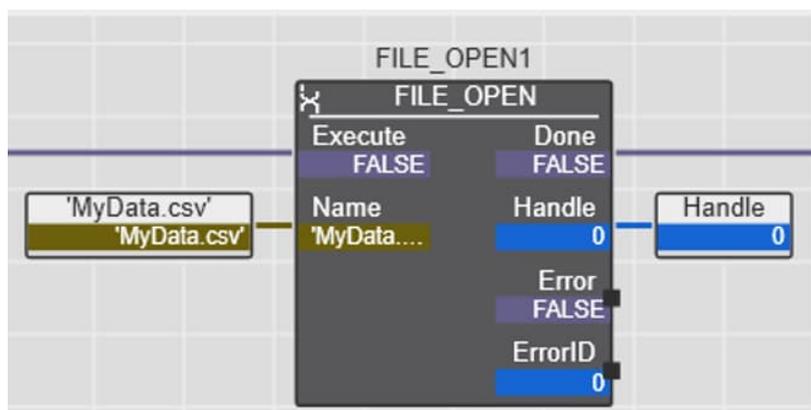
BUF_CNTよりも多い配列数を指定。ここでは整数配列33個以上を用意。

DONE- bool型: ファイルバッファ成功(TRUE) 次ページに続く

プログラム例

- ファイルオープン処理を行う

COMPONENTSのSearch画面でFILE_OPENを入力し、FILE_OPENのファンクションブロックをドラック&ドロップしてください

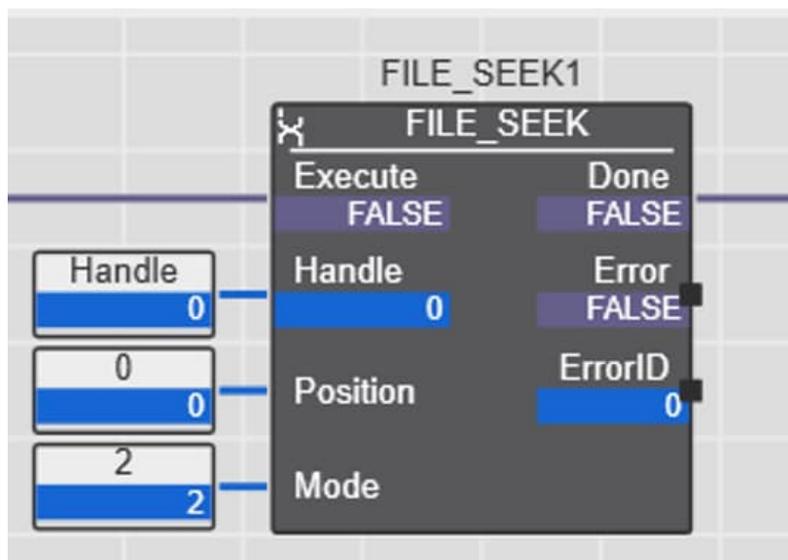


- Execute- bool型: ここでは前ページのSTRING_TO_BUFのDoneを結合する
TRUEでCSVファイルへ出力したいファイルを開く
- Name- 文字列型: 変数名(ここではMyData.csv)
書き込みたいファイル名を指定する(ファイルがない場合は新規作成される)
- Done - bool型: ファイルオープン成功(TRUE) 次ページに続く
- Handle- 整数型: 開かれたファイルが新規作成(0)か、
既存ファイルを開いたか(1)

プログラム例

- 書き込みたいファイルのファイルポインタを指定する(新規で上書きするか、末尾に追加するか)

COMPONENTSのSearch画面でFILE_SEEKを入力し、FILE_SEEKのファンクションブロックをドラック&ドロップしてください

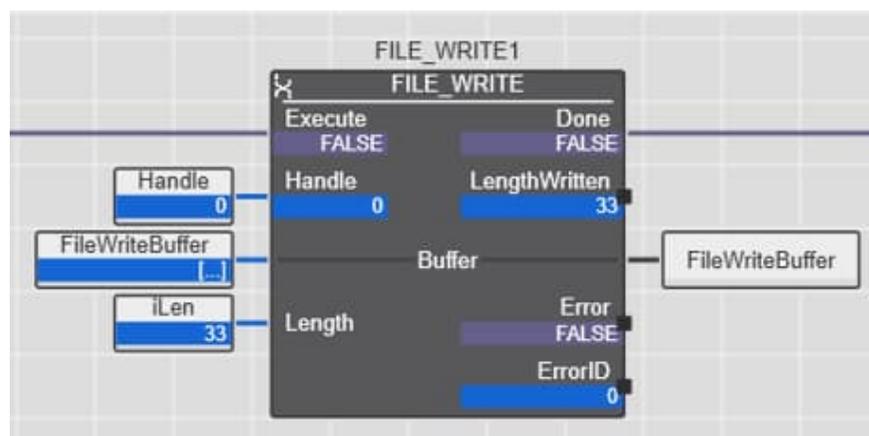


- Execute- bool型:ここでは前ページのFILE_OPENのDONEを結合する
TRUEでCSVファイルへ出力したいファイルを開く
- Name- 文字列型:変数名(ここではMyData.csv)
書き込みたいファイル名を指定する(ファイルがない場合は新規作成される)
- Done - bool型:ファイルポインタの実行に成功(TRUE) 次ページに続く
- Handle- 整数型:開かれたファイルが新規作成(0)か、
既存ファイルを開いたか(1)
- Mode- 整数型:ここでは2(ファイルの末尾に追加出力する)

プログラム例

- ファイルへの書き込みを実行する

COMPONENTSのSearch画面でFILE_WRITEを入力し、FILE_WRITEのファンクションブロックをドラック & ドロップしてください

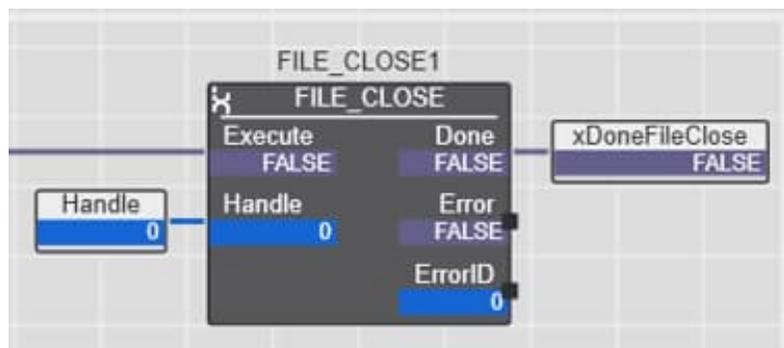


- Execute- bool型: ここでは前ページのFILE_SEEKのDoneを結合する
TRUEでCSVファイルへ出力したいファイルを開く
- Length- 整数型: 変数名(ここではiLen)
一度の実行でバッファから書き込みたい文字列のサイズを指定
- Done - bool型: ファイルへの書き込みに成功(TRUE) 次ページに続く
- Handle- 整数型: 変数名(ここではHandle)
開かれたファイルが新規作成(0)か、既存ファイルを開いたか(1)
- Buffer- 整数型配列: バッファ格納先変数(ここではFileWriteBuffer)

プログラム例

- ファイルクローズ処理を実行する

COMPONENTSのSearch画面でFILE_CLOSEを入力し、FILE_CLOSEのファンクションブロックをドラック & ドロップしてください

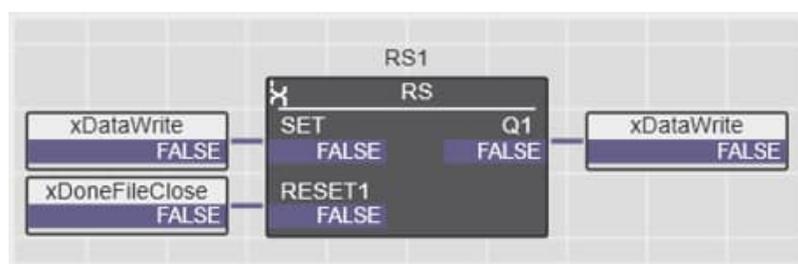


- Execute- bool型: ここでは前ページのFILE_WRITEのDoneを結合する
TRUEでCSVファイルへ出力したファイルをクローズする
- Length- 整数型: 変数名(ここではiLen)
一度の実行でバッファから書き込みたい文字列のサイズを指定
- Done - bool型: ファイルクローズに成功(TRUE) ここではxDoneFileClose変数に書き込み
- Handle- 整数型: 変数名(ここではHandle)
開かれたファイルが新規作成(0)か、既存ファイルをオープンしたか(1)

プログラム例

- ファイルクローズ処理のタイミングでファイル書き込み処理を終了する

COMPONENTSのSearch画面でRSを入力し、RSのファンクションブロックをドラック&ドロップしてください



SET – 変数名(ここではxDataWrite)

TRUEでQ1に設定された変数をTRUEにする

RESET1 – 変数名(ここではxDoneFileClose)

TRUEでQ1に設定された変数をFALSEにする

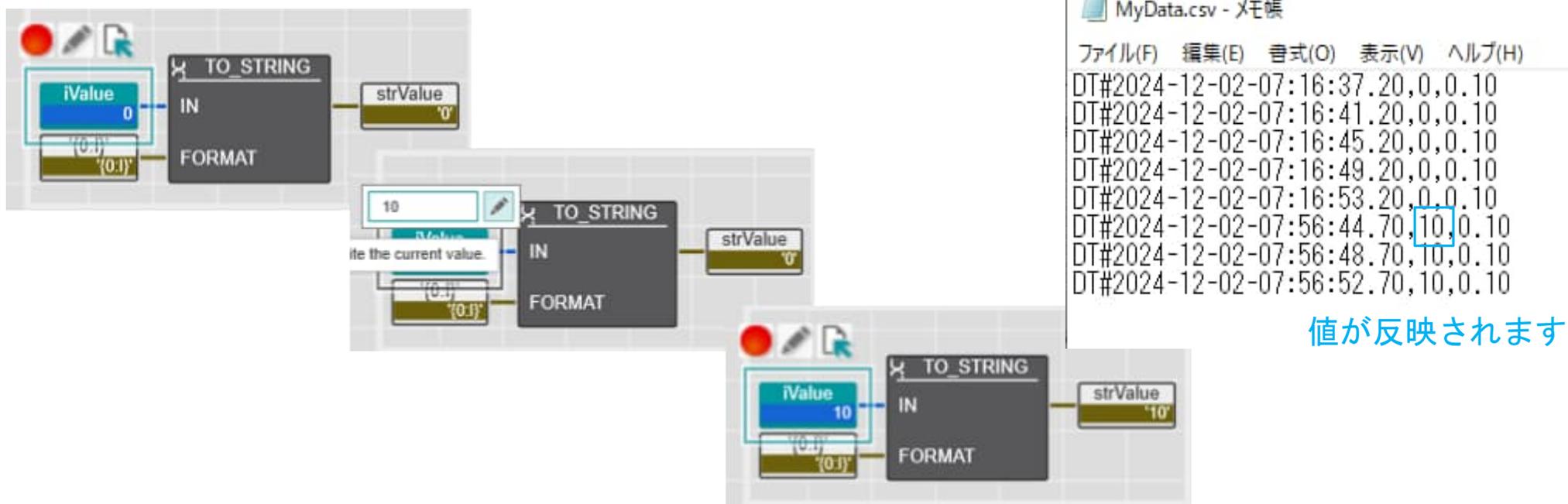
Q1 – 変数名(ここではSETと同じxDataWrite)

RESET1の状態に合わせてをFALSEにする

プログラム例

- 出力したい値を任意に変更してみる

PLCnext EngineerよりPLCnext ControlへアクセスしDebugモードでiValueの値を変更する



MyData.csv - メモ帳

ファイル(F)	編集(E)	書式(O)	表示(V)	ヘルプ(H)
DT#2024-12-02-07:16:37.20	,0	,0	,0.10	
DT#2024-12-02-07:16:41.20	,0	,0	,0.10	
DT#2024-12-02-07:16:45.20	,0	,0	,0.10	
DT#2024-12-02-07:16:49.20	,0	,0	,0.10	
DT#2024-12-02-07:16:53.20	,0	,0	,0.10	
DT#2024-12-02-07:56:44.70	,10	,0	,0.10	
DT#2024-12-02-07:56:48.70	,10	,0	,0.10	
DT#2024-12-02-07:56:52.70	,10	,0	,0.10	

値が反映されます

出力先

- プログラムで出力されるファイルはデフォルトでは/opt/plcnextディレクトリに出力されます

名前	サイズ	更新日時	パーミッション	所有者
..		2023/06/09 20:20:44	rw-rw-r--	admin
apps		2019/12/03 19:28:42	rw-rw-r--	plcnext...
appshome		2019/12/03 19:28:42	rw-rw-r--	plcnext...
config		2021/09/13 23:01:56	rw-rw-r--	admin
data		2021/09/13 23:02:55	rw-rw-r--	plcnext...
installed_apps		2019/12/03 19:28:23	rw-r--r--	root
logs		2024/12/02 16:37:23	rw-rw-r--	plcnext...
ltnng		2018/03/09 21:34:56	rw-rw-r--	admin
profinet		2021/09/13 17:59:55	rw-rw-r--	plcnext...
projects		2019/12/03 19:28:24	rw-rw-r--	admin
retaining		2021/09/13 23:02:47	rw-rw-r--	admin
Security		2019/12/03 19:27:46	rw-rw-r--	admin
shadowing		2018/03/09 21:34:56	rw-rw-r--	admin
MyData.csv	4 KB	2024/12/02 16:56:54	rw-rw-r--	plcnext...

ディレクトリを指定する場合はFILE_OPENファンクションブロックのNameでパスを指定することができます

PLCnext Control内のディレクトリにアクセスする場合はPCにWinSCPをインストールしていただき、アクセスしてください

[PLCnext へのファイル転送 | PLCnext Technology](#)

参考(変数リスト)

- 参考として今回使用したデモでの変数リストは下記です。

RTC_S1		RTC_S	Local	<input type="checkbox"/>		
RS1		RS	Local	<input type="checkbox"/>		
STRING_TO_BUF1		STRING_TO_BUF	Local	<input type="checkbox"/>		
FILE_OPEN1		FILE_OPEN	Local	<input type="checkbox"/>		
FILE_SEEK1		FILE_SEEK	Local	<input type="checkbox"/>		
FILE_WRITE1		FILE_WRITE	Local	<input type="checkbox"/>		
FILE_CLOSE1		FILE_CLOSE	Local	<input type="checkbox"/>		
xDoneFileClose	FALSE	BOOL	Local	<input type="checkbox"/>		FALSE
Handle	0	UINT	Local	<input type="checkbox"/>		UINT#0
FileWriteBuffer	[...]	ARRAY41_OF_INT	Local	<input type="checkbox"/>		
iLen	33	INT	Local	<input type="checkbox"/>		INT#0
Q	TRUE	BOOL	Local	<input type="checkbox"/>		FALSE
strTime	'DT#2024-12-02-...	STRING	Local	<input type="checkbox"/>		*
strSeperator	','	STRING	Local	<input type="checkbox"/>		*
strValue	'0'	STRING	Local	<input type="checkbox"/>		*
strRow	'DT#2024-12-02-...	STRING	Local	<input type="checkbox"/>		*
iValue	0	INT	Local	<input type="checkbox"/>		INT#0
xDataWrite	FALSE	BOOL	Local	<input type="checkbox"/>		FALSE
LoadCurrent_Ch1_Real	0.1	REAL	External	<input type="checkbox"/>		
strLoadCurrentCh1	'0.10'	STRING	Local	<input type="checkbox"/>		*

Thank you