株式会社マイクロネット
鹿島開発センター
〒314-0135 茨城県神栖市堀割 3-8-11
TEL 0299-90-1733　FAX 0299-92-8557

作成日：2023/03/13
作成者：小柳　正久

CSV ファイル読み込みサンプル

1. 概要

この資料は RT-C 言語コントローラで CSV ファイルからデータを共有メモリへ読込んだデータを配列として参照するサンプルです。

1)このサンプルでは「C:¥¥Work¥¥RTC_Test.csv」に配置した CSV ファイルデータを読み込みます。
2)CSV データは 10 列構成としており、読込んだ文字列を int 型に変換してメモリに保管します。
3)メモリ上に保管されたデータを２次元配列として読み込むサンプルです。

2. C#プログラム

2.1. RTC_Program_Common.cs

1)　コンストラクタでは共有メモリを確保して。
2)　通常処理で読み込んだデータを 2 次元配列として取り扱える関数[Convert2Drray()]を通してデバッグ用の変数へ値を代入しています。

```
using Eclr;
using Eclr.Pcos;
using ProConOS_eCLR;
using RTC_APILib;
using RTC_CAN_Lib;
using RTC_COMLib;
using RTC_ECAT_Lib;
using RTC_LOGLib;
using RTC_SOCKLib;
using System;
using System.Iec61131Lib;


namespace RTC_Template
{
    /// <summary>
    /// RT-C共通処理クラス
    /// </summary>
    ///

    public unsafe class CRTC
    {
        //-----------------------------------
        //■ステータス定義
        //-----------------------------------
        public const short STS_INI = 0;    //初期処理
        public const short STS_PRO = 1;    //通常処理
        public const short STS_END = 2;    //終了処理
        public const short STS_ERR = 3;    //エラー処理

        //-----------------------------------
        //■構造体定義
        //-----------------------------------

        //デバッグ用構造体（デバッグ時下記の構造体とデバッグツールの構造体を合わせる必要が有ります）
        public struct M_AREA
        {
            public int inSystemCounter; //システム時間カウンタ、システム時間毎に増加
            //-------------------------------------------------------
            //↓以下、ユーザが構造体定義を記述します。
            //-------------------------------------------------------
            public int inC_MajorVersion;    //メジャーバージョン
            public int inC_MinorVersion;    //マイナーバージョン
            public int inC_BuildVersion;    //ビルドバージョン
```

株式会社マイクロネット
鹿島開発センター
〒314-0135 茨城県神栖市堀割 3-8-11
TEL 0299-90-1733　FAX 0299-92-8557

```
    public int Array_0_0;           //サンプル　2次元配列データ[0][0]
    public int Array_9_9;           //サンプル　2次元配列データ[9][9]
    //-------------------------------------------------------
    //↑ユーザが記述する構造体定義はここまでです。
    //-------------------------------------------------------
}

//-----------------------------------
//■変数定義
//-----------------------------------

//M_AREA共有メモリポインタ変数
public static M_AREA* pMArea;

//メモリ取得
public static RtMemory rmData;  //共有メモリ
public static int      *pData;  //ポインタ変数

/// <summary>
/// コンストラクタ
/// </summary>
static CRTC()
{
    //-----------------------------------
    //■デバッグ用メモリポインタ取得
    //-----------------------------------

    //デバッグ用メモリポインタ取得
    pMArea = (M_AREA*)cSharedArea.GetMemPtr();
    pMArea->inSystemCounter = 0;    //システムカウンタ

    //-----------------------------------
    //C#プログラムバージョン
    //-----------------------------------
    pMArea->inC_MajorVersion = 1;   //メジャーバージョン
    pMArea->inC_MinorVersion = 2;   //マイナーバージョン
    pMArea->inC_BuildVersion = 3;   //ビルドバージョン

    //-----------------------------------
    //■共有メモリ取得
    //-----------------------------------
    IecString80 sMemName = new IecString80();
    sMemName.ctor();
    sMemName.s.Init("RTC_Data");    //共有メモリ名（最大12文字）

    uint SharedMemSize = 4096 * 10; //共有メモリサイズ(4096Byte単位)　例：4096 * 10 = 40KB

    rmData = new RtMemory(SharedMemSize, (byte)HandleSelection.ROOT_PROCESS, sMemName);  //メモリ領域を確保
    if (rmData.GetLastErr() != 0)
    {
        //既に共有メモリが作成済みの場合
        IecString80 sProcName = new IecString80();
        sProcName.ctor();
        sProcName.s.Init("");

        rmData = new RtMemory(sProcName, sMemName); //既にある共有メモリを使用
        if (rmData.GetLastErr() != 0)
        {
            //共有メモリ取得エラー
            return;
        }
    }

    pData = (int *) rmData.GetMemPtr(); //共有メモリポインタ取得
}
```

読込んだデータを RTC 開発・デバッグツール（MULTIPROG）を使用しオンラインでデバッグ出来るように構造体メンバーに追加しています。

CSV ファイルから読込んだデータを保管する共有メモリ用の変数宣言を追加

CSV ファイルから読込んだデータを保管する共有メモリ用の変数宣言を追加

共有メモリ用にメモリを確保し、ポインタを取得

確保する共有メモリサイズを指定します。
サイズは 4096byte の倍数を指定してください。

承認	照査	作成	資料名称	図番	No.
		小柳 2023.03.13	RT-C 言語コントローラ　CSV ファイル読込と 2 次元配列サンプル	[MF0080]	2／6

株式会社マイクロネット
鹿島開発センター
〒314-0135 茨城県神栖市堀割 3-8-11
TEL 0299-90-1733　FAX 0299-92-8557

```
        /// <summary>
        /// デストラクタ
        /// </summary>
        ~CRTC()
        {
            //共有メモリが有効でであれば解放
            if (rmData != null)
            {
                rmData.Delete();     //データ格納領域を解放する
                rmData = null;       //変数初期化
            }
        }
    }


}
```

確保したメモリを解放します。

3. RTC_Program_01.cs

1) 初期処理で CSV ファイルからデータを読み込み共有メモリの配列に保管しています。
2) 通常処理で読み込んだデータを 2 次元配列として取り扱える関数[Convert2Drray()]を通してデバッグ用の変数へ値を代入しています。

```
using Eclr;
using Eclr.Pcos;
using ProConOS_eCLR;
using RTC_APILib;
using RTC_CAN_Lib;
using RTC_COMLib;
using RTC_ECAT_Lib;
using RTC_LOGLib;
using RTC_SOCKLib;
using System;
using System.Iec61131Lib;


namespace RTC_Template
{

    [FUNCTION("INT")]
    public unsafe class RTC_Program_01
    {
        public static  int arraySize = 10;   //CSVデータの列数（サンプル：CSV項目数10列）

        /// <summary>
        /// １次元の共有メモリの配列を２次元配列へ変換
        /// </summary>
        /// <param name="Arr1"></param>
        /// <param name="Arr2"></param>
        /// <param name="Array2Size"></param>
        /// <returns></returns>
        unsafe private static int Convert2Drray(int Arr1,int Arr2,  int Array2Size)
        {
            return CRTC.pData[(Arr1 * Array2Size) + Arr2];
        }


        /// <summary>
        /// Task01で動作するプログラム
        /// </summary>
        /// <param name="inState">ステータス定義(STS_INI:初期処理、STS_PRO:通常処理、STS_END:終了処理、
STS_ERR:エラー処理)</param>
        /// <returns></returns>
```

CSV データの列数を指定します。サンプルでは 10 列としています。

1 次元配列を２次元配列として扱える関数を用意しました。

株式会社マイクロネット
鹿島開発センター
〒314-0135 茨城県神栖市堀割 3-8-11
TEL 0299-90-1733　FAX 0299-92-8557

```
        public static Int16 __Process([VAR_INPUT("INT")] Int16 inState)
        {
            Int16 outState = 99;                // 終了時のステータス格納用変数

            switch (inState)
            {
                case CRTC.STS_INI:
                    outState = On_Init();       //初期処理
                    break;
                case CRTC.STS_PRO:
                    outState = On_Process();    //通常処理
                    break;
                case CRTC.STS_END:
                    outState = On_End();        //終了処理
                    break;
                case CRTC.STS_ERR:
                    outState = On_Err();        //エラー処理
                    break;
            }

            return outState;                    // FUの実行結果を返却
        }

        //-----------------------------------
        //■ 初期処理
        //-----------------------------------
        static short On_Init()
        {
            Int16 inReturn = CRTC.STS_INI;

            //-------------------------------------------------------------
            //↓以下、ユーザが初期処理を記述します。
            //-------------------------------------------------------------

            //---------------------------------
            //CSVファイル読込
            //---------------------------------

            int iLine = 0;                                      //読み込み行数
            int dataLen = 0;                                    //読み込みデータサイズ
            bool bAns;                                          //指定文字列を指定デリミタで分割した結果

            IecString80 buffe = new IecString80();              //ファイルデータ読み込みバッファ
            buffe.ctor();
            buffe.s.Init("");

            IecString80 sFmt = new IecString80();               //データフォーマット文字
            sFmt.ctor();
            sFmt.s.Init("%d");

            IecString80[] str = new IecString80[arraySize];     //デリミタで分割した文字列データ

            FileAccesser sMemName = new FileAccesser("C:¥¥Work¥¥RTC_Test.csv"); //読み込みファイルパスセット


            //ファイルオープン
            if (sMemName.Open() == false)
            {
                //ファイルオープンエラー
            }

            //ファイル全行読み込み
            while (true)
            {
                //1行読込
                dataLen = sMemName.ReadLine(ref buffe);
                if(dataLen <= 0)
                {
```

読込む CSV ファイルのパスを指定してください。

株式会社マイクロネット
鹿島開発センター
〒314-0135 茨城県神栖市堀割 3-8-11
TEL 0299-90-1733　FAX 0299-92-8557

```
                //読み込みデータなし
                break;
            }

            //デリミタで分割
            bAns = sMemName.Spilit(buffe, 0x2c, ref str, arraySize);
            if (bAns)
            {
                //分解エラー
            }

            //共有メモリへ書き込み
            for (int r = 0; r < arraySize; r++)
            {
                //文字列から指定フォーマットに変換し共有メモリへ書き込み
                CRTC.pData[(iLine * arraySize) + r] = Helper.StoI(str[r].s.GetIecString(),
sFmt.s.GetIecString());
            }

            iLine += 1;      //読み込み行数インクリメント
        }

        //ファイルクローズ
        sMemName.Close();

        inReturn = CRTC.STS_PRO;     //通常処理へ状態移行
        //------------------------------------------------------
        //↑ユーザが記述する初期処理はここまでです。
        //------------------------------------------------------

        return inReturn;
    }

    //---------------------------------
    // ■通常処理
    //---------------------------------
    static short On_Process()
    {
        Int16 inReturn = CRTC.STS_PRO;

        //------------------------------------------------------
        //↓以下、ユーザが通常処理を記述します。
        //------------------------------------------------------

        //２次元配列データ取得サンプル
        //※設定・デバッグツール（MULTIPROG）のウォッチウィンドウで確認出来ます。
        CRTC.pMArea->Array_0_0 = Convert2Drray(0, 0, arraySize); //2次元配列データ[0][0]
        CRTC.pMArea->Array_9_9 = Convert2Drray(9, 9, arraySize); //2次元配列データ[9][9]

        //------------------------------------------------------
        //↑ユーザが記述する通常処理はここまでです。
        //------------------------------------------------------

        return inReturn;
    }
```

ファイルから読込んだデータをデリミタ(「,」=0x2c)ごとに文字列に分割します！

文字列から int 型に変換

1 次元配列を２次元配列として扱える関数を通してデータを取得しデバッグ用の変数へ代入

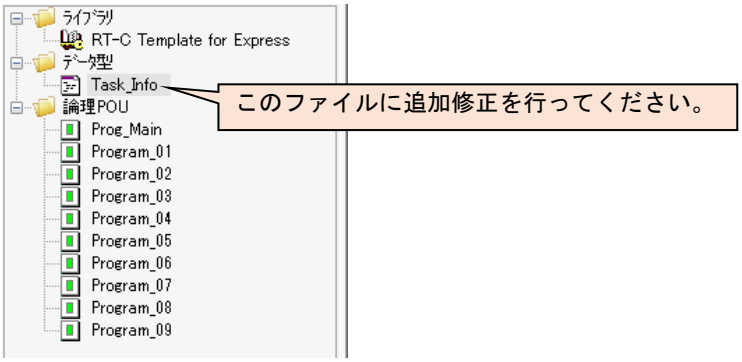
3.1. FileAccess.cs

ファイルアクセス用のクラスファイルです。
プロジェクトへ追加してご利用ください。

株式会社**マイクロネット**
鹿島開発センター
〒314-0135 茨城県神栖市堀割 3-8-11
TEL 0299-90-1733　FAX 0299-92-8557

4. MULTIPROG

RTC 開発・デバッグツール（MULTIPROG）でデータ型にデバッグ用の構造体メンバーを登録しウォッチウィンドで確認出来るようにします。




```
TYPE
    M_AREA:
 STRUCT
        inSystemCounter          : DINT;
        (* ↓以下、ユーザが記述する構造体定義を定義します。 *)

        inC_MajorVersion  : DINT; (*メジャーバージョン*)
        inC_MinorVersion  : DINT; (*マイナーバージョン*)
        inC_BuildVersion  : DINT; (*ビルドバージョン*)

        inArray_0_0    :DINT;       (*サンプル　2 次元配列データ[0][0]*)
        inArray_9_9   :DINT;       (*サンプル　2 次元配列データ[9][9]*)

        (* ↑ユーザが記述する構造体定義はここまでです。 *)
 END_STRUCT;
END_TYPE
```

このサンプルでは 2 次元配列の[0][0]と[9][9]の値をウォッチウィンドを使用しオンラインで確認出来るようにしています。

以上

承 認	照 査	作 成	資 料 名 称	図 番	No.
		小柳 2023.03.13	RT-C 言語コントローラ　CSV ファイル読込と 2 次元配列サンプル	[MF0080]	6／6