

Руководство пользователя

Описание интерфейса библиотеки ФБ ArchivatorOwenLib для ведения архива

Дата редакции 23.12.10
Версия 1

«Овен»

Описание и назначение библиотеки

Данная библиотека предназначена для ведения архива данных и записи его в энергонезависимую память контроллера (внутреннюю или внешнюю). Для этого используется два ФБ: первый блок «GetData» формирует строку для записи, на него заводятся те переменные, которые мы хотим архивировать; второй блок «WriteData» осуществляет запись сформированной строки в архив, в перспективе осуществляет несколько режимов ведения архива (пока доступен только один: каждый день создается новый файл).

Описание ФБ

Блок «GetData»

Входы:

- Name - имя переменной, максимальная длина 15 символов, используется при формировании заголовка архива, если включен режим формирования заголовка (об этом будет сказано подробнее ниже)
- Tip – тип записываемой переменной: 0-REAL, 1-Byte, 2-Word, 3-DWord, 4-String (максимальная длина такая же, как у имени – 15 символов)
- Znachenie - значение переменной

Выходы:

- ErrCode -код ошибки (255-переполнение строки данных, 254-переполнение заголовка, 253-переполнение и строки данных и заголовка, 0-ошибок нет); если ошибка случилось, то блок продолжает работать, только новые данные не будут добавлены в конечную строку данных или заголовка.
- StrOut – сформированная строка данных определенного формата (номер переменной в виде #xxx и значение в шестнадцатичной форме записи, например #005=00A1, «#»Номер переменной ПРОБЕЛ «=»Значение переменной в зависимости от типа)
- StrOutHead –сформированная строка заголовка определенного формата («#»Номер переменной ПРОБЕЛ «size=»Размер в байтах ПРОБЕЛ«name=» имя переменнойПРОБЕЛ)

Внутренние методы и функции:

- АКЦИОН «Clean» - очистка внутренних переменных ФБ, выполняется перед повторным формированием строки, вызывается в смежном блоке «WriteData» из этой же библиотеке
- АКЦИОН «SetHead» - при вызове разрешает формирование строки заголовка(т.к. в связи с форматом заголовка и строки данных, первый получается гораздо длиннее, а длина строки для обоих одинакова и **равна 255 символов**, то иногда целесообразно не создавать заголовка); заголовок формируется один раз за первый цикл и далее не меняется

Блок «WriteData»

Входы:

- Enable – разрешение на работу
- ArcPnt – ссылка на ФБ формирования заголовка и строки данных «GetData»
- NameArc – имя архива(максимум 15 символов)
- Path – путь к папке, в которой будет храниться архив (*в конце должна стоять косая черта, например: /var/log/*)

Для записи данных на флеш-карту или SD-карту памяти у контролеров ПЛК304/308 путь выглядит следующим образом:

SD: « /mnt/mmc/»

USB: «/mnt/usb1(2)/»

- Mode – режим работы (0-каждый день создается новый файл архива, дата добавляется в имя архива)

Выходы:

- ErrCode –код ошибки (0-ошибок нет, 1-не удалось создать файл, 2-не удалось открыть или файл занят, 3- не удалось записать), если ошибка случилось, то происходит выход из ФБ

Внутренние методы и функции:

- ACTION «CloseArc» - закрывает файл архива (для более красивого кода)
- METHOD «CreateArc» - создает новый архив, при успешном завершение выдает «TRUE»
- METHOD «IsBeArc» - проверяет на существование файла архива, если существует выдает «TRUE»
- METHOD «OpenArc» - открывает файл архива, возвращает ссылку на открытый файл(SysFile.RTS_IEC_HANDLE)
- METHOD «WriteInArc» - запись данных в файл архива, в случае успеха возвращает TRUE

Все они используются внутри самого ФБ, поэтому более подробно их описывать не будем.

Пример использования

Список переменных:

```
PROGRAM PLC_PRG
```

```
VAR
```

```
  Arc1_1:GetData; //блок формирования строки и заголовка
```

```
  Arc1_2:WriteData; // блок записи в архив
```

```
  strT:STRING:='bla bla bla'; //некоторая текстовая переменная
```

```
  in1:BYTE; //некоторые внутренние переменные
```

```
  in2: WORD;
```

```
  in3:DWORD;
```

```
  b:BOOL; //вспомогательная переменная
```

```
  tp1:ton; // таймер для задания периода архивации
```

```
END_VAR
```

Сама программа:

```
b:=TRUE; //задаем ее каждый цикл TRUE для работы таймера
```

```
in1:=in1+1; // некоторые операции с данными
```

```
in2:=in2+2;
```

```
in3:=in3+123;
```

```
IF tp1.Q THEN
```

```
  b:=FALSE; // обеспечит перезапуск таймера(его циклическую работу)
```

END_IF

Arc1_1.SetHead(); //разрешаем составить заголовок (обязательно ставить перед объявление переменных для архивирования)

Arc1_1(Name:='in1', Tip:=1, Znachenie:=ADR(in1)); // записываем значение переменной «in1»

Arc1_1(Name:='in2', Tip:=2, Znachenie:=ADR(in2));

Arc1_1(Name:=left('in3 name bolee 15 sim',15), Tip:=3, Znachenie:=ADR(in3)); // имя переменной обязательно должно быть меньше 16 символов, если больше, то может вызвать ошибку

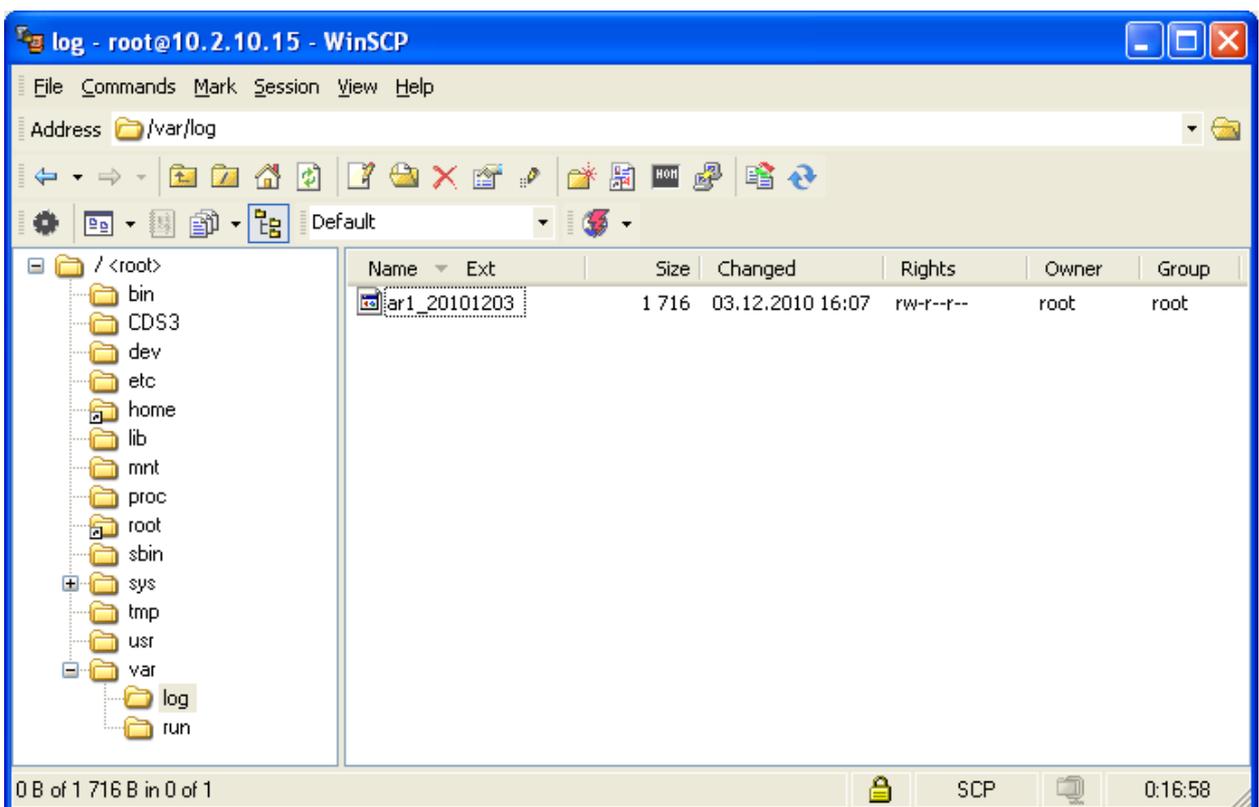
Arc1_1(Name:='text', Tip:=4, Znachenie:=ADR(strT));

//запишем все эти данные в архив, разрешение на работу свяжем с таймером, данные берем из ФБ «Arc1_1», имя архива будет «ar1», и хранится будет в папке «/var/log/»(путь надо завершать косой чертой), режим оставляем по умолчанию равны «0»

Arc1_2(Enable:=tp1.Q, ArcPnt:=adr(Arc1_1), NameArc:='ar1', Path:='/var/log/');

TP1(IN := b, PT:= T#1M); //запустим таймер

На ПЛК получим файл(ы) следующего вида



Как видно к имени архива добавилась дата.

Со следующим содержанием:

//заголовок

#000 size=001 name=in1 #001 size=002 name=in2 #002 size=004 name=in3 name bolee #003 size=015 name=text

//данные

2010.12.03 16.05.22 #000=66 #001=00CC #002=00003102 #003=bla bla bla
2010.12.03 16.05.27 #000=CC #001=0198 #002=00006204 #003=bla bla bla
2010.12.03 16.05.32 #000=32 #001=0264 #002=00009306 #003=bla bla bla
2010.12.03 16.05.37 #000=98 #001=0330 #002=0000C408 #003=bla bla bla
2010.12.03 16.05.42 #000=FE #001=03FC #002=0000F50A #003=bla bla bla
2010.12.03 16.05.47 #000=64 #001=04C8 #002=0001260C #003=bla bla bla

2010.12.03 16.05.52 #000=CA #001=0594 #002=0001570E #003=bla bla bla
2010.12.03 16.05.57 #000=30 #001=0660 #002=00018810 #003=bla bla bla
2010.12.03 16.06.02 #000=96 #001=072C #002=0001B912 #003=bla bla bla
2010.12.03 16.06.07 #000=FC #001=07F8 #002=0001EA14 #003=bla bla bla
2010.12.03 16.06.13 #000=62 #001=08C4 #002=00021B16 #003=bla bla bla
2010.12.03 16.06.18 #000=C8 #001=0990 #002=00024C18 #003=bla bla bla
2010.12.03 16.06.23 #000=2E #001=0A5C #002=00027D1A #003=bla bla bla

И Т.Д.