# Миграция проектов DirectSOFT в Do-more Designer

Утилита ***Migrate DirectSOFT Project*** помогает программисту в перемещении проекта **DirectSOFT**, написанного для контроллеров PLC в проект **Do-more Designer,** который будет использован в ПЛК Do-more.

Перед началом перемещения проекта откройте утилиту: **Do-more Designer-> File-> Import-> Migrate DirectSOFT Project** (Эта утилита входит в состав Do-more Designer версии 1.3)

Эта утилита предназначена для ПЕРЕМЕЩЕНИЯ (*migration),* а не ПРЕОБРАЗОВАНИЯ проекта, так как некоторые элементы программирования проектов DirectSOFT не существуют или не требуются в ПЛК Do-more.

Многие элементы проекта DirectSOFT (контакты, обмотки, таймеры, счетчики, сдвиговые регистры и др.) будут перемещены в проект Do-more Designer с минимальными усилиями (обычно, требуется только корректировка адресов памяти и привязки к входам/выходам).

Но другие элементы не имеют эквивалента в Do-more, потому что они обрабатываются по-разному (например: математические операции в DirectSOFT выполняются, как последовательность различных арифметических команд, а в Do-more есть единственная команда MATH для выполнения всех арифметических действий).

Кроме того в проекте DirectSOFT существуют элементы, которые совсем не нужны в проекте Do-more (например, релейная логика для настройки 'пойнтерным методом' считывания сигналов аналоговых модулей).

Любые элементы, которые не могут быть обработаны автоматически, будут перенесены в проект Do-more в форме команд **DirectLOGIC Stub** (**$DL, $DL1**, **$DLBLK**) – т.е. КОМАНД-ЗАГЛУШЕК.
Их необходимо вручную заменить на подходящие команды Do-more и, затем, удалить из проекта Do-more, до загрузки проекта в ЦПУ Do-more.

Привязки к вводу/выводу и памяти (*I/O and memory references*), которые не имеют эквивалентов в ПЛК Do-more CPU, будут перемещены в ***Unassigned Nicknames***. Привязки, также, должны быть заменены и, затем, *Unassigned Nicknames* должны быть удалены из проекта Do-more.

По умолчанию, все тексты элемента (*DL Element* и *Nickname - Documentation*) проекта DirectSOFT перемещаются в конец 6-ти строчных полей *Description* – Описания, проекта Do-more.

Исходным проектом может быть проект DirectSOFT32 (версия 4) или DirectSOFT5 для ЦПУ:
D0-05, D0-06, D2-230, D2-240, D2-250, D2-250-1, D2-260, D3-350, D4-430, D4-440, D4-450.

Целевым проектом может быть проект для любого ЦПУ Do-more CPUs:
Simulator (DM-SIM), H2-DM1, H2-DM1E, T1H-DM1, T1H-DM1E

В процессе миграции используется проект DirectSOFT, экспортированный в текстовый файл.

**Создание экспортного файла проекта DirectSOFT:**

- ☐ В DirectSOFT Programming откройте нужный проект.
- ☐ Выберите в меню ***File-> Export-> Program***. Нажмите ***OK***.

**ПРИМЕЧАНИЕ:** ***НЕ ДЕЛАЙТЕ*** *пометку в окошке «**Expand I-Boxes» -** Развернуть I-Box**, в** окне **«Export Options».***

После завершения экспорта, закройте DirectSOFT и откройте утилиту **Migrate DirectSOFT Project**

**Откройте экспортный файл проекта DirectSOFT:**

Щелкните по кнопке ***Browse*,** найдите и выделите файл проекта. Затем нажмите «***Open»***. Затем нажимайте кнопку «**Далее»,** которые последовательно открывают 5 окон с напоминанием особенностей перемещения отдельных элементов проекта и возможностями выбора, связанных с вариантами переадресации (перепривязки) - *Remapping Options.*

## Операции с переадресацией ввода/вывода - *Handling I/O Module Mapping*

**Дискретные модули - *Discrete I/O Module Mapping***

ЦПУ DirectLOGIC и ЦПУ Do-more используют память **X** и **Y** для модулей *Discrete Input* и *Discrete Output*, память X и Y в ПЛК DirectLOGIC пронумерована в восьмеричной (**octal**) системе, тогда, как по умолчанию в ПЛК Do-more нумерация десятичная (*decimal*).

Утилита миграции предлагает на выбор три варианта:

1. **Map DirectLOGIC X/Y (octal) to Do-more X/Y (decimal), (default)** – Просто заменить октальные адреса X/Y на децимальные эквиваленты. Существующие ***Nicknames, Wiring Info, Descriptions*** будут следовать за элементом.

   *Примеры перемещения:*
   DirectLOGIC X0 будет отображен в Do-more X0
   ...
   DirectLOGIC X7 будет отображен в Do-more X7
   DirectLOGIC X10 будет отображен в Do-more X8
   DirectLOGIC X11 будет отображен в Do-more X9
   ...
   DirectLOGIC X377 будет отображен в Do-more X255

2. **Map DirectLOGIC X/Y (octal) to Do-more DLX/DLY (octal)** – Заменить октальные адреса X/Y на октальные адреса DLX /DLY. В ПЛК Do-more блоки памяти DLX и DLY изначально предназначены для обмена данными с внешними устройствами по протоколу K-Sequence. Существующие ***Nicknames, Wiring Info, Descriptions*** будут следовать за элементом.

   *Примеры перемещения:*
   DirectLOGIC X0 будет отображен в Do-more DLX0
   ...
   DirectLOGIC X7 будет отображен в Do-more DLX7
   DirectLOGIC X10 будет отображен в Do-more DLX10
   ...
   DirectLOGIC X377 будет отображен в Do-more DLX377

3. **Map DirectLOGIC X/Y (octal) to Unassigned Nicknames** - Заменить октальные адреса X/Y на НЕ ПРИВЯЗАННЫЕ к физическим входам/выходам ячейкам *Unassigned Nicknames*, которые могут быть использованы для привязки к любым X и Y. Привязки, должны быть осуществлены и, затем, *Unassigned Nicknames* должны быть удалены из проекта Do-more.

   - ☐ Если ссылка на X или Y в проекте DirectLOGIC имеет Nickname, ТОЛЬКО **Nickname** (и если есть, Wiring Info, Descriptions) будет сохранено, ссылки на элементы НЕ будут перенесены
   - ☐ Если ссылка на X или Y в проекте DirectLOGIC не имеют Nickname, ***Unassigned Nickname*** будет создано со ссылками на X или Y сдобавлением спереди "***DL***_".

   *Примеры перемещения:*
   Все ссылки на X0 будут перенаправлены к Nickname "DL_X0"
   Все ссылки на X7 будут перенаправлены к Unassigned Nickname "DL_X7"
   ...
   Все ссылки на X10 будут перенаправлены к Unassigned Nickname "DL_X10"
   Все ссылки на X11 будут перенаправлены к Unassigned Nickname "DL_X11"

   Все ссылки на X100 с Nickname '**Start_Motor**' будут перенаправлены к ***Unassigned Nickname 'Start_Motor',*** а в поле Description будет добавлено '**X100**'

## **Аналоговые модули - *Analog I/O Module Mapping***

Сигналы аналоговых модулей ПЛК DirectLOGIC отображаются изначально в память *Discrete I/O* . Затем их с помощью команд релейной логики размещают в *V-Memory*. В ПЛК Do-more этого не нужно делать, поэтому эту часть логики надо удалить.

CPUs do not require any ladder logic to handle the analog I/O modules.

- ☐ Сигналы аналоговых входов ПЛК Do-more автоматически отображаются в **WX memory** (и, по желанию, сигналы диагностики в *X memory*). *WX memory* - ***Signed Word*** (16 бит).
- ☐ Выходы аналоговых модулей автоматически отображаются в ***WY memory*** (и, по желанию, настройки в *Y memory*). WY memory - ***Signed Word*** (16 бит).

Так как аналоговые сигналы, обычно, требуют масштабирования, то релейная логика проекта DirectLOGIC может быть заменена командами ***Scale Value (SCALE)***. Поместите команды SCALE для Analog Input Modules в Системный блок - **$tTopOfScan** System Code block, а команды SCALE дляг Analog Output Modules в Системный блок - **$tBottomOfScan** System Code block.

## **Промежуточные реле - *Handling Control Relay Mapping***

ЦПУ DirectLOGIC и ЦПУ Do-more используют ***C memory*** для Control Relays – Промежуточных реле. C memory в ПЛК DirectLOGIC пронумерована в ***octal***, в ПЛК Do-more по умолчанию в ***decimal.***

1. **Map DirectLOGIC Control Relay (octal) to Do-more Control Relay (decimal)** – Замена октальных адресов C на эсвивалентные десятичные адреса. Существующие ***Nicknames, Wiring Info, Descriptions*** будут следовать за элементом.

   *Примеры перемещения:*
   DirectLOGIC C0 будет отображен в Do-more C0
   ...
   DirectLOGIC C7 будет отображен в Do-more C7
   DirectLOGIC C10 будет отображен в Do-more C8
   DirectLOGIC C11 будет отображен в Do-more C9
   ...
   DirectLOGIC C377 будет отображен в Do-more C255

2. **Map DirectLOGIC C (octal) to Do-more DLC (octal)** - Заменить октальные адреса C на октальные адреса DLC. В ПЛК Do-more блоки памяти DLC изначально предназначены для обмена данными с внешними устройствами по протоколу K-Sequence. Существующие ***Nicknames, Wiring Info, Descriptions*** будут следовать за элементом.

   *Примеры перемещения:*
   DirectLOGIC C0 будет отображен в Do-more DLC0
   ...
   DirectLOGIC C7 будет отображен в Do-more DLC7
   DirectLOGIC C10 будет отображен в Do-more DLC10
   ...
   DirectLOGIC C377 будет отображен в Do-more DLC377

3. **Map DirectLOGIC C (octal) to Unassigned Nicknames** - Заменить октальные адреса C на НЕ ПРИВЯЗАННЫЕ *Unassigned Nicknames*, которые могут быть использованы для привязки к любым C. Привязки, должны быть осуществлены и, затем, *Unassigned Nicknames* должны быть удалены из проекта Do-more.
   - ☐ Если ссылка на C в проекте DirectLOGIC имеет Nickname, ТОЛЬКО **Nickname** (и если есть, Wiring Info, Descriptions) будет сохранено, ссылки на элемент НЕ будут перенесены
   - ☐ Если ссылка на C в проекте DirectLOGIC не имеет Nickname, ***Unassigned Nickname*** будет создано со ссылками на X или Y с добавлением спереди "***DL***_".

   *Примеры перемещения:*
   Все ссылки на C0 будут перенаправлены к Unassigned Nickname "DL_C0"
   Все ссылки на C7 будут перенаправлены к Unassigned Nickname "DL_C7"
   ...
   Все ссылки на C10 будут перенаправлены к Unassigned Nickname "DL_C10"

Все ссылки на C11 будут перенаправлены к Unassigned Nickname "DL_C11"

Все ссылки на C100 с *Nickname* 'Start_Motor' будут перенаправлены к *Unassigned Nickname* 'Start_Motor' и в поле *Description* будет добавлено 'C100'

**<u>Память пользователя V-memory - *Handling User V Memory Mapping*</u>**

ЦПУ DirectLOGIC и ЦПУ Do-more используют память общего назначения: ***V-Memory***, ***Unsigned Word (16-бит)***. Адреса V-Memory в ПЛК DirectLOGIC октальные, а в ПЛК Do-more, по умолчанию, децимальные.

У ПЛК DirectLOGIC есть дополнительно еще области памяти: ***System V-Memory, Timer/Counter memory,*** и т. д., которая используется для настроек и информации о состоянии. Обращения к к этим областям V-Memory не может быть перемещено, потому что нет соответствующих функций в ПЛК Do-more. Ссылки на эти области памяти всегда замещаются на ***Unassigned Nicknames***, и все команды с обращением к этим областям должны быть удалены.

1. **Map DirectLOGIC User V (octal) to Do-more V (decimal)** – Замена всех октальных адресов V-памяти на их десятичные эквиваленты. Существующие ***Nicknames, Wiring Info, Descriptions*** будут следовать за элементом.

   *Примеры перемещения:*
   DirectLOGIC V2000 будет отображен в Do-more V1024
   ...
   DirectLOGIC V2007 будет отображен в Do-more V1031
   DirectLOGIC V2010 будет отображен в Do-more V1032.
   ...
   DirectLOGIC V37777 будет отображен в Do-more V16383

2. **Map DirectLOGIC V (octal) to Do-more DLV (octal)** - Заменить октальные адреса V на октальные адреса DLV. В ПЛК Do-more блоки памяти DLV изначально предназначены для обмена данными с внешними устройствами по протоколу K-Sequence. Существующие ***Nicknames, Wiring Info, Descriptions*** будут следовать за элементом .

   *Примеры перемещения:*
   DirectLOGIC V2000 будет отображен в Do-more DLV2000
   ...
   DirectLOGIC V2007 будет отображен в Do-more DLV2007
   DirectLOGIC V2010 будет отображен в Do-more DLV2010
   .....
   DirectLOGIC V37777 будет отображен в Do-more DLV37777

3. **Map DirectLOGIC V (octal) to Unassigned Nicknames** - Заменить октальные адреса V на НЕ ПРИВЯЗАННЫЕ *Unassigned Nicknames*, которые могут быть использованы для привязки к любым V. Привязки, должны быть осуществлены и, затем, *Unassigned Nicknames* должны быть удалены из проекта Do-more.

   - ☐ Если ссылка на V в проекте DirectLOGIC имеет Nickname, ТОЛЬКО **Nickname** (и если есть, Wiring Info, Descriptions) будет сохранено, ссылки на элемент НЕ будут перенесены
   - ☐ Если ссылка на V в проекте DirectLOGIC не имеет Nickname, ***Unassigned Nickname*** будет создано со ссылками на X или Y с добавлением спереди "***DL***_"

   .*Примеры перемещения:*
   Все ссылки на V0 будут перенаправлены к Unassigned Nickname "DL_V0"
   Все ссылки на V7 будут перенаправлены к Unassigned Nickname "DL_V7"
   ...
   Все ссылки на V10 будут перенаправлены к Unassigned Nickname "DL_V10"
   Все ссылки на V11 будут перенаправлены к Unassigned Nickname "DL_V11"

   Все ссылки на V100 с *Nickname* 'Start_Motor' будут перенаправлены к Unassigned Nickname 'Start_Motor' и поле *Description* будет дополнено 'V100'

**Таймеры и Счетчики - *Handling Timer and Counter Mapping***

Таймеры ПЛК DirectLOGIC имеют октальные адреса T0 - T377 и используют адреса V0 - V377 для хранения 16-ти битовых аккумулированных значений. Аккумулирующие Таймеры используют по два последовательных адреса V-Memory в диапазоне V0 - V777для хранения 32-х битовых аккумулированных значений.

Счетчики ПЛК DirectLOGIC имеют октальные адреса CT0 - CT377 и используют адреса V1000 – V1377 для хранения 16-ти битовых аккумулированных значений. «*Up/Down*» счетчики используют по два последовательных адреса V-Memory в диапазоне V1000 – V1777для хранения 32-х битовых аккумулированных значений

Таймеры и Счетчики ПЛК Do-more используют структуры, с соответствующими полями для хранения данных о состоянии.

Контакты, привязанные к *Timer / Counter,* будут замещены, как **<Timer>.Done** или **<Counter>.Done.**

Сравнительные контакты, которые ссылаются на *Timer* или *Counter* будут перенаправлены к **<Timer>.Acc** или **<Counter>.Acc.**

Другие ссылки к текущему значению Таймера (V0 - V377) будут перенаправлены к **<Timer>.Acc**

Другие ссылки к текущему значению Счетчика (V1000 - V1377) будут перенаправлены к **<Counter>.Acc**

Все Таймеры ПЛК Do-more используют миллисекундное разрешение (нет - 0.1 сек. или 'fast - 0.01 сек.' вариантов Таймеров). *Timers* и *Fast Timers* будут перемещены с структуры Timer. Значение константы **Timer Preset** DirectLOGIC будет преобразовано в миллисекунды. Любые другие ссылки на **Timer Preset** должны быть изменены вручную.

1. **Map DirectLOGIC User T/CT (octal) to Do-more Timer/Counter (decimal)** - Просто заменить октальные адреса T и CT на децимальные эквиваленты. Существующие ***Nicknames, Wiring Info, Descriptions*** будут следовать за элементом.

   *Примеры перемещения:*
   DirectLOGIC T0 будет отображен в Do-more T0
   ...
   DirectLOGIC T7 будет отображен в Do-more T7
   DirectLOGIC T10 будет отображен в Do-more T8
   .....
   DirectLOGIC T377 будет отображен в Do-more T255

2. **Map DirectLOGIC T/CT (octal) to Unassigned Nicknames** – Заменить октальные адреса T или CT на НЕ ПРИВЯЗАННЫЕ *Unassigned Nicknames*, которые могут быть использованы для привязки к любым T или CT. Привязки, должны быть осуществлены и, затем, *Unassigned Nicknames* должны быть удалены из проекта Do-more.

   - ☐ Если ссылка на T или CT в проекте DirectLOGIC имеет Nickname, ТОЛЬКО **Nickname** (и если есть, Wiring Info, Descriptions) будет сохранено, ссылки на элементы НЕ будут перенесены
   - ☐ Если ссылка на T или CT в проекте DirectLOGIC не имеют Nickname, ***Unassigned Nickname*** будет создано со ссылками на X или Y с добавлением спереди "***DL***_"

   *Примеры перемещения:*
   Все ссылки на T0 будут перенаправлены к Unassigned Nickname "DL_T0"
   Все ссылки на T7 будут перенаправлены к Unassigned Nickname "DL_T7"
   ...
   Все ссылки на T10 будут перенаправлены к Unassigned Nickname "DL_T10"
   Все ссылки на T11 будут перенаправлены к Unassigned Nickname "DL_T11"

   Все ссылки на T100 с *Nickname* 'Motor_On_Time' будут перенаправлены к Unassigned Nickname 'Motor_ON_Time' и в поле *Description* будет добавлено '_T100'

**Биты Стадий - *Handling Stage Bit Mapping***

ЦПУ DirectLOGIC и ЦПУ Do-more используют S-memory дл Стадий (*Stages*). ПЛК DirectLOGIC может содержать до 1024 команды Стадий, пронумерованных от S0 до S1777.

Каждый блок-кода PROGRAM в ПЛК Do-more может содержать до 128 команд Стадий, которые имеют десятичную адресацию: от **<CodeBlockName>.0** до **<CodeBlockName>.127.**

1. **Map DirectLOGIC Stage (octal) to Do-more Stage (decimal)** – Замена Стадий пронумерованных от S0 до S177 на их децимальные эквиваленты (S0 - S127) в ***$Main code block***. Стадии, пронумерованные от S200 и далее, перемещаются, как *Unassigned Nicknames*. Существующие ***Nicknames, Wiring Info, Descriptions*** будут следовать за элементом Стадия.
   *Примеры перемещения:*
   DirectLOGIC S0 будет отображен в Do-more $Main.S0
   .....
   DirectLOGIC S177 будет отображен в Do-more $Main.S127
   ...
   DirectLOGIC S200 будет отображен в Do-more Unassigned Nickname "DL_S200"
   DirectLOGIC S1777 будет отображен в Do-more Unassigned Nickname "DL_S1777"

2. **Map DirectLOGIC Stage (octal) to Unassigned Nicknames** - Заменить октальные адреса Стадий на НЕ ПРИВЯЗАННЫЕ *Unassigned Nicknames*, которые могут быть использованы для привязки к любым Стадиям. Привязки, должны быть осуществлены и, затем, *Unassigned Nicknames* должны быть удалены из проекта Do-more.

   - ☐ Если ссылка на Стадию в проекте DirectLOGIC имеет Nickname, ТОЛЬКО **Nickname** (и если есть, Wiring Info, Descriptions) будет сохранено, ссылки на элементы НЕ будут перенесены
   - ☐ Если ссылка Стадию в проекте DirectLOGIC не имеют Nickname, ***Unassigned Nickname*** будет создано со ссылками на X или Y с добавлением спереди "***DL***_"

   *Примеры перемещения:*
   Все ссылки на S0 будут перенаправлены к Unassigned Nickname "DL_S0"
   Все ссылки на S7 будут перенаправлены к Unassigned Nickname "DL_S7"
   ...
   Все ссылки на о S10 будут перенаправлены к Unassigned Nickname "DL_S10"
   Все ссылки на S11 будут перенаправлены к Unassigned Nickname "DL_S11"

   Все ссылки на S100 с *Nickname* 'Motor_Control' будут перенаправлены к Unassigned Nickname 'Motor_Control' и поле *Description* будет дополнено 'S100'

**Выбор целевого ЦПУ Do-more - *Select the Target Do-more CPU***

Используйте окна «***Do-more Hardware Class*»** и «***Type*»** для выбора целевого ЦПУ Do-more.

**ПРИМЕЧАНИЕ:**Тип ЦПУ в Do-more Designer можно всегда изменить, используя ***Setup Offline CPU***.

***Migration Process Log File* – Файл журнала процесса перемещения**

**Generate migration log file and open it once completed –** По умолчанию, в процессе перемещения создается файл - log file (с названием **<Project_Name>_DoMore_Log.Txt** в папке, где размещен DirectSOFT *export file*). В этом файле содержится вся информация о процессе генерации, которая будет использована в оставшейся части процесса миграции.

***Merge similar log items* – Объединить одинаковые элементы при отображении в файле сообщений**

Типичный проект DirectSOFT содержит много копий команд и привязок к памяти, которые не могут быть автоматически перемещены в эквивалентные команды и ссылки Do-more. Каждая копия (*instance*) будет отмечена в процессе миграции и, в зависимости от этого выбора, будет формироваться файл сообщений:

**Enabled (default)** – Создается одна запись в файле независимо от числа копий команд и ссылок в проекте DirectSOFT project.

**Disabled** - Создается запись в файле для каждой копии команды или ссылки.

**Просмотр выбранного варианта - *Review Migration Selections***

В этом окне отображены все настройки, сделанные на предыдущих этапах. Вернитесь назад, для внесения изменений или нажмите кнопку «***Finish***».

Операция миграции проекта это многопроходовый процесс.

При обработке каждого элемента проекта сообщения о состоянии или об ошибках будут отображаться в окне ***Output***.

О ходе процесса миграции создается журнал сообщений **<Export_Filename>_DoMore_Log.Txt**. В этом файле будут сохранена вся информация, появляющаяся в окне *Output* Window во время процесса миграции.

При завершении процесса миграции, появится окно сообщений «***First Migration Step Successful***» - Первый шаг миграции проекта успешно завершен. Нажмите **OK, чтобы увидеть созданный проект** Do-more Designer. После открытия проекта появится сообщение «***Open Log File***».

**Ручное завершение перемещения - *Manually Completing the Migration***

Автоматическая часть процесса миграции создает допустимый (*valid)* проект Do-more Designer, который содержит цепи релейной логики и элементы программы. Но процесс миграции, обычно, не в состоянии автоматически обработать все элементы проекта DirectSOFT, и это означает, что необходимо ручное редактирование созданного проекта, для завершения процесса миграции.

Далее будут приведены советы по возможно быстрому ручному редактированию проекта. Ручное редактирование – это итеративный процесс, в котором Вы просматриваете все сообщения в ***log file***. Для редактирования нужно иметь:

- ☐ Исходный проект DirectSOFT
- ☐ Файл <Export Filename>_DoMore_Log.Txt
- ☐ Созданный автоматически проект Do-more Designer
- ☐ Если есть ПЛК Do-more, соединитесь с ним.

Если нет ПЛК Do-more, используйте ***System Configuration-> I/O Configuration-> Manual configuration*** для ручного выбора и настройки модулей ПЛК. В результате Вы получите привязку физических входов/выходов - ***I/O Mapping.***

***Complete the I/O Module Mapping* – Завершение привязки модулей ввода/вывода**

**Discrete Input / Output Modules**

- ☐ Помните, что ПЛК Do-more используют десятичные адреса ***I/O***, а не октальные.
- ☐ Файл помощи, тема: DMD0229 ***Mapping for Discrete Input and Output Modules***
- ☐ Все ссылки на ***GX, GY*** будут направлены к *Unassigned Nicknames* и должны быть заменены или удалены из проекта, до загрузки в ПЛК.

**Analog Input / Output Modules**

- ☐ Удалите все цепи, связанные с настройкой ввода аналоговых входов/выходов – они не нужны в Do-more. ПЛК автоматически отображает их в памяти WX/ WY..
- ☐ Файл помощи, тема: DMD0226: ***Mapping for Analog Input and Output Modules***. В нем также отмечены модули, для которых требуется начальные настройка - *Module Configurations*.
- ☐ Цепи масштабирования аналоговых входов можно легко преобразовать, используя команду ***Scale Value (SCALE)*** instructions.
- ☐ Обратите внимание на цепи обработки ошибок ввода (обрыв и т.п.). В файле помощи можно узнать, как эти ошибки обрабатываются в Do-more и сделайте соответствующие изменения в программе.

**CTRIO / CTRIO2 Modules**

- ☐ Do-more Designer не использует для настройки этих модулей ***CTRIO Workbench***. Они настраиваются в части ***CTRIO Module Configuration*** функции ***System Configuration.***
- ☐ ***Module Configuration*** может импортировать существующую «*CTRIO configuration*» из ***CTRIO Workbench*** в виде файла (**.CWB**).
- ☐ Любые команды или I-Box используемые для управления содулями CTRIO необходимо заменить на команды ***Do-more CTRIO instructions***.
- ☐ Используйте функцию ПЛК: ***PLC-> Monitor CTRIO Module*** для взаимодействия с модулями CTRIO/2 в режиме работы.

**Complete the Communication with External Devices – соединения с внешними устройствами**

- ☐ ПЛК Do-more используют «***device driver***» для настройки встроенных портов и портов коммуникационных модулей.
- ☐ Коммуникационные команды **I-Box** DirectSOFT обычно хорошо мигрируют, но релейная логика с командами "LD / LD / LDA / RX " іне может мигрировать, как одна коммуникационная команда Do-more.
- ☐ ПЛК DirectLOGIC используют Специальные реле 'Port Busy' SP для блокировки команд RX или WX. Логика для блокировки не нужна в командах Do-more, потому что это реализуют «***device driver***» самостоятельно.
- ☐ ПЛК Do-more используют **Modbus** и **K-Sequence Server** для связи с устройствами-клиентами, использующими протоколы Modbus и K-Sequence. Команды Read / Write от

этих устройств имеют доступ только для областей памяти, выделенных в ПЛК Do-для каждого протокола.

- Внешние Modbus /TCP и Modbus /RTU клиенты будут иметь доступ только к блокам памяти: **MI (Modbus Input), MC (Modbus Coil), MIR (Modbus Input Register), MHR (Modbus Holding Register)**.
- Внешние K-Sequence клиенты будут иметь доступ только к блокам памяти: DLX (DirectLOGIC Input), DLY (DirectLOGIC Output), DLC (DirectLOGIC Control Relay), DLV (DirectLOGIC Register).

**ПРИМЕЧАНИЕ:** Используйте файл помощи, тема DMD0206: ***Built-in Data Blocks***

***Resolve $DL, $DL1, and $DLBLK (DirectLOGIC Stub) instructions* – Замена команд-заглушек**

Все команды $DL, $DL1 , $DLBLK должны быть вручную заменены. Их нужно заменить соответствующими командами Do-more и, затем, удалены из проекта. Эти команды будут содержать три строки (***string entries***):

- Первая строка - мнемоника команды DirectSOFT
- Вторая строка представляет одно из трех:
  - Команду Do-more с подобными функциями с описанием
  - **"n/a"**, если нет эквивалентной по функциям команды Do-more
  - **"n/a",** если не требуется в ПЛК Do-more
- Третья строка содержит дополнительную информацию о том, как использовать содержимое этой команды-заглушки - *DirectLOGIC Stub*.

---

Большинство DL SP (Special Relay) не заменяются Do-more ST (Status) битами, потому что их функции не требуются в ПЛК Do-more. Они перенаправляются к ***Unassigned Nicknames*** которые должны быть заменены или удалены из проекта.
**ПРИМЕЧАНИЕ:** Файл помощи, тема DMD0208: ***System Nicknamed Bit Locations*** содержит список ***System-Assigned Nickname*** ПЛК Do-more

---=

Все обращения к ячейкам ***System Variable Memory*** перенаправляются к ***Unassigned Nicknames*** потому что их функции не требуются в ПЛК Do-more. Они должны быть заменены или удалены из проекта.
**ПРИМЕЧАНИЕ:** Файл помощи, тема DMD0208: ***System Nicknamed DWord Locations*** содержит список ячеек **DWord** в Do-more, у которых есть ***System-Assigned Nickname***.

---

Арифметические операции (Add, Subtract, Multiply, Divide, и др.) в ПЛК DirectLOGIC выполняются последовательными цепочками команд, а ПЛК Do-more используют единственную команду ***MATH - Calculate Expression*** для всех арифметических действий.

Обычный формат чисел в ПЛК DirectLOGIC используемый в командах (Add, Subtract, Multiply, Divide, и др.) - это BCD. ПЛК Do-more используют форматы: ***decimal*** (т.е. *binary*) и ***Real*** по умолчаниют. Необходимо проверить все цепи использующие математические команды, что бы получить корректную замену команд.

---

Do-more использует ссылки к массивам (*array references*) для косвенной адресации – например, **D[ V0 ],** где V0 содержит «index number» (или указатель, пойнтер) от 0 до 65535. Для косвенной адресации в ПЛК DirectLOGIC используются Пойнтеры - **P memory** – например, P2000, где содержимое V2000 интрепретируется, как **Octal V-Memory address**. Все ссылки на **P memory** будут перенаправлены к *Unassigned Nicknames* и должны быть заменены или удалены из проекта.

---

ПЛК Do-more использует структуры ***Date/Time*** и команды Date/Time для работы с ними, тогда как ПЛК DirectLOGIC используют системную память **System V-Memory** для хранения компонентов часов/календаря. Цепи использующие эти системные переменные должны быть заменены с использованием структуры **$Now** Date/Time (Сейчас) и командами Date Time. Все ссылки на эти системные переменные будут перенаправлены к *Unassigned Nicknames* и должны быть вручную заменены или удалены из проекта..

**ПРИМЕЧАНИЕ:** Файл помощи, тема DMD0222: ***Managing Date and Time in a Do-more CPUs*** содержит описание структуры Date/Time и команд Date/Time.

**<u>Преобразование проекта - *Convert the Project to Do-more*</u>**

Преобразование проекта DirectLOGIC в проект Do-more не только перемещение элементов релейной логики, привязки к модулям и обращений к памяти. Конверсия проекта подразумевает перезапись частей релейной логики, чтобы получить преимущества от использования свойств ПЛК Do-more..

Используйте преимущества встроенных модулей Do-mores перемещая части релейной логики во встроенные системные Задачи - ***built-in System Tasks,*** например:

- ☐ Цепи, использующие **SP0 (FirstScan),** могут быть перенесены в **$tFirstScan System Task**. Рассмотрите возможность использования команд: **Set Range (SETR) / Reset Range (RSTR), Initialize Data (INIT),** если в проекте DirectLOGIC были использованы команды LD & OUT ідля ввода начальных значений на первом скане ПЛК.
- ☐ Цепи масштабирования аналоговых входов могут быть помещены в **$tTopOfScan System Task**
- ☐ Цепи масштабирования аналоговых входов могут быть помещены в **$tBottomOfScan System Task**

Используйте преимущества модульности, перемещая части релейной логики в пользовательские блоки кода: ***Program*** или ***Task code blocks***. Вот некоторые примеры:

- ☐ Если проект содержит части логики, которая выполняется по условию (через интервалы времени или событию), рассмотрите возможность перемещения этих частей логики в отдельные Задачи - ***Tasks*** с добавлением команды **Enable Task (ENTASK)** для каждого созданного блока кода - *Task code-bock*.
- ☐ Если в проекте есть цепи, обрабатываемые асинхронно, или обработка цепей занимает несколько скан-циклов ПЛК, рассмотрите возможность переместить эти цепи в блоки кода типа Программа - **Program** и добавление команды **Run Program (RUN)** для каждой созданной программы.
- ☐ Если проект содержит несколько схем со Стадиями (*Stage*) рассмотрите возможность переместить эти Стадии в отдельные Программы (+ команда **Run Program (RUN)**) і. Также рассмотрите возможность помещения каждой команды DirectLOGIC **ISG (Initial Stage)** в новую Программу. (У ПЛК Do-more нет команды ISG, но первая стадия в каждой программе является активной при пуске этой Программы).
- ☐ Если в проекте есть части релейной логики, которую Вы хотите защитить (read-only with password to unlock), переместите эту логику в отдельную Программу, затем используйте один из вариантов защиты - ***Code-Block Protection***. (Смотри файл помощи, тема: DMD0231).

---

Если в ПЛК DirectLOGIC используется ERM с EBC100 или GS-EDRV100, то стоит рассмотреть вариант удаления модуля ERM и использования вместо него функции встроенного порта Do-more - ***Ethernet I/O Master.*** Основное преимущество состоит в том, что ЦПУ Do-more взаимодействует с модулями Ввода/вывода в EBC100 также, как и с модулями в локальном каркасе.