Марафон **Marathon**

Сценарный CANopen конфигуратор

Руководство пользователя

Код проекта: **1000**$_h$

Москва, 2011

Оглавление

Введение.

Сценарный CANopen конфигуратор предоставляет возможность исполнения наборов команд, которые осуществляют взаимодействие с различными CANopen устройствами в соответствии со стандартом CiA 301 v. 4.2. Сценарий представляет собой текстовый файл определенного формата. Сценарный конфигуратор распространяется в виде дополнительного модуля (DLL Plugin) для программы CANwise версий 3.4 и выше. Документация подготовлена с использованием пакетов OpenOffice и LibreOffice.

Соглашения по документации.

Принятые сокращения.

CiA	Международная организация CAN in Automation - "CAN в автоматизации".
CAN-ID	Идентификатор CAN кадра канального уровня.
COB-ID	Идентификатор коммуникационного объекта CANopen.
NMT	Сетевой менеджер: определяет объекты управления CANopen сетью.
PDO	Объект данных процесса; обеспечивает обмен компактными данными (до 8 байт) в режиме жесткого реального времени.
RTR	Удаленный запрос объекта.
SDO	Сервисный объект данных; обеспечивает обмен большими объемами данных в режиме мягкого реального времени.
M	Обязательный (mandatory) объект.
O	НЕ обязательный (optional) объект.
LSB	Наименее значимый (младший) бит или байт.
MSB	Наиболее значимый (старший) бит или байт.
RO	Доступ только по чтению.
WO	Доступ только по записи.
RW	Доступ по чтению и записи.
RWR	Доступ по чтению и записи, асинхронный доступ по чтению (для PDO) .
RWW	Доступ по чтению и записи, асинхронный доступ по записи (для PDO) .

Для подробного ознакомления с терминологией рекомендуется использовать CAN словарь, изданный на русском языке организацией CAN in Automation (Москва, 2005). Электронная версия словаря размещена здесь.

Обозначение основных типов данных.

boolean	Логическое значение true/false.
int8	Целое 8 бит со знаком.
unsigned8	Беззнаковое целое 8 бит.
int16	Целое 16 бит со знаком.
unsigned16	Беззнаковое целое 16 бит.
int32	Целое 32 бита со знаком.
unsigned32	Беззнаковое целое 32 бита.
int64	Целое 64 бита со знаком.
unsigned64	Беззнаковое целое 64 бита.
real32	32-х разрядное с плавающей точкой.
real64	64-х разрядное с плавающей точкой.
vis-string	Строка видимых ASCII символов (коды 0 и 20h - 7Eh).

Изменения в версиях.

Версия 1.2.0

В сценарный конфигуратор интегрирован анализатор траффика CANopen протокола. Предоставляется возможность записи результатов анализа в файл протокола выполнения сценария. См. новые управляющие операторы [AnalyzerOn] и [AnalyzerOff]. Версия файла сценария 10000102.

Версия 1.3.0

При выборе имен файлов сценария и протокола используется обновленное диалоговое окно.

Версия 1.4.0.

Обеспечена поддержка версии 4.2 стандарта DS301 от 07 декабря 2007 г. Для этого введены два типа SYNC кадров с длиной поля данных 0 и 1 байт. Версия файла сценария 10000103.

Документация сценарного CANopen конфигуратора отредактирована и переведена в формат pdf.

Версия 1.5.0

Используется механизм сохранения конфигурации сценарного CANopen конфигуратора между запусками (сессиями). Поддерживается программой CANwise версий 3.4 и выше.

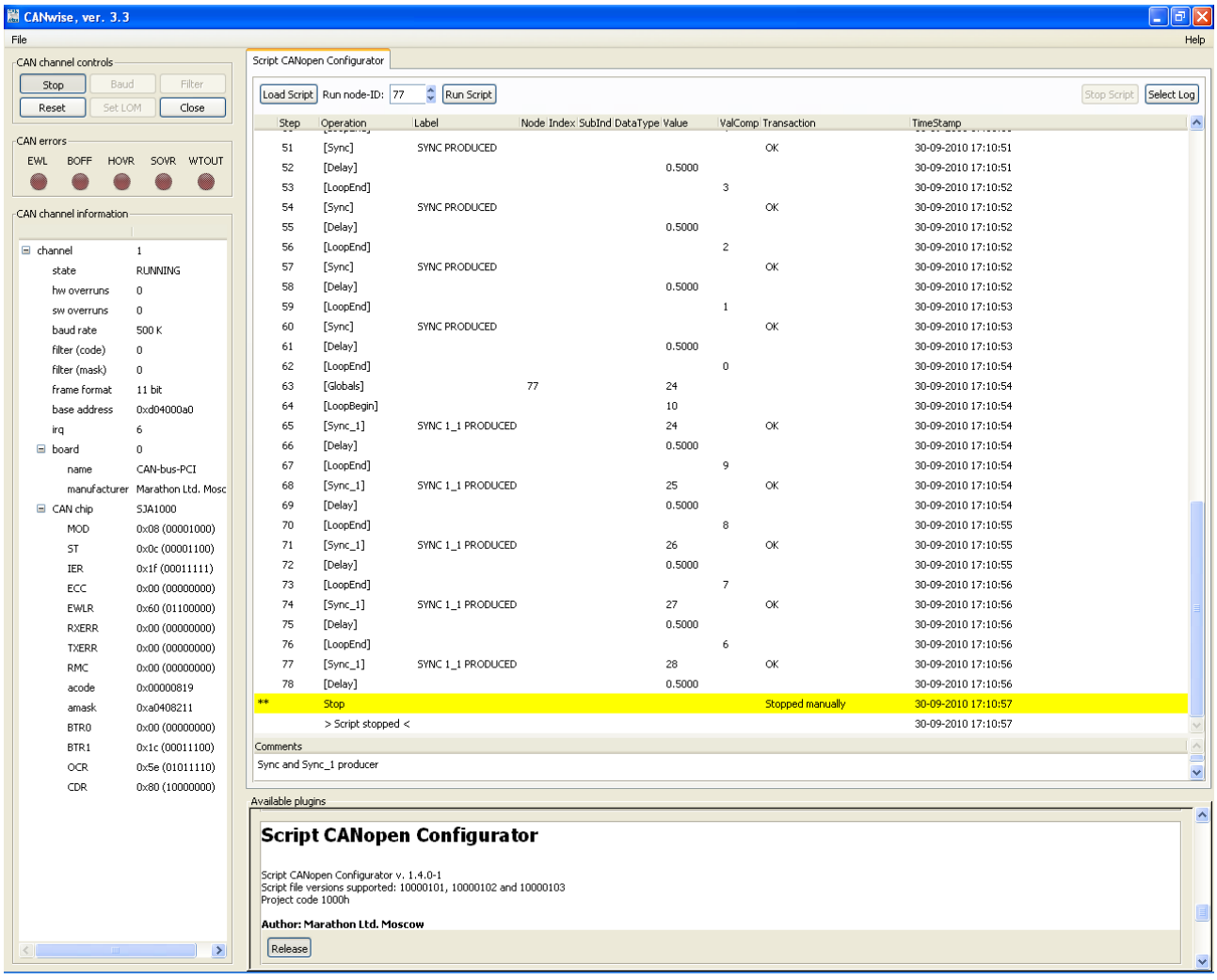
Установка и запуск программы.

Установка программы.

1. Установить драйвер CHAI канального уровня CAN сети в соответствии с инструкциями, размещенными на сайте http://can.marathon.ru/page/prog/chai.
2. Установить программу CANwise по инструкции, размещенной на сайте http://can.marathon.ru/page/prog/canwise.
3. Установить интерактивный CANopen конфигуратор путем записи модуля CANopenScript.dll в корневую директорию программы CANwise.

Запуск программы конфигуратора.

Модуль конфигуратора содержит два прикладных окна и набор кнопок общего управления:



Для начала работы с CANwise нужно выполнить следующие операции:

- При необходимости задать скорость CAN сети (по умолчанию устанавливается скорость 500 кбит/с);
- Запустить CANwise кнопкой Start;

Назначение кнопок общего управления.

Кнопка	Назначение
Load script	Загружает файл сценария (расширение по умолчанию psc) и осуществляет его компиляцию.
Run node-ID	Номер CAN узла, передаваемый в сценарий при запуске (0..127). При исполнении сценария номер CAN узла является глобальной переменной (далее обозначается как *node-ID*). Если оператор сценария не содержит инструкции, устанавливающей новое значение *node-ID*, используется текущее значение глобальной переменной.
Run script	Запуск сценария на выполнение. Возможен только в случае, если сценарий скомпилирован без ошибок. Сообщения об ошибках заносятся в нижнее окно конфигуратора.
Stop script	Ручной останов выполнения сценария.
Select log	Выбор имени файла для протокола выполнения сценария. По умолчанию используется имя файла сценария с расширением slg. Запись протокола в файл является обязательной.

Структура и операторы сценария.

Сценарий представляет собой текстовый файл составленный из набора операторов. Оператор в свою очередь может содержать несколько обязательных и опционных полей. Имя оператора заключается в прямоугольные скобки и размещается в отдельной строке. Каждое поле оператора также должно размещаться в отдельной строке. Последовательность полей может быть любой. В имени оператора и названиях полей строчные и прописные буквы не различаются.

Возможно использование одного или нескольких форматирующих символов (табуляция, пробел) в начале любой строки. При отделении названия и значения полей операторов обязательно присутствие хотя бы одного форматирующего символа.

Все целочисленные значения могут записываться в десятичном, восьмеричном (начинается с нуля) или шестнадцатеричном (начинается с 0X или 0x) виде. Логические значения (тип BOOLEAN) задаются в виде True или False.

В файл сценария могут быть внесены комментарии. Комментарием считается любая строка либо часть строки, следующая за символами // (два символа / с наклоном вправо). Для комментирования большого сегмента текста, состоящего из нескольких строк, используются символы /* при открытии комментария и */ при его закрытии. Весь закомментированный текст пропускается и не анализируется компилятором сценария. Пустые строки, либо состоящие только из символов форматирования, также игнорируются.

В нижеследующем описании имена операторов и названия обязательных полей выделены жирным шрифтом. Пример сценария приведен в файле **Sample_Script.psc**

Описательные операторы.

Выполняют описательные функции в файле либо алгоритме сценария, при этом не влияют на ход исполнения алгоритма.

[PSCR 10000103]

Задает номер версии файла сценария. В данном случае это 10000103. Оператор версии всегда должен быть первым оператором сценария.

[Comments]

Оператор комментария.
Содержит произвольный текст, состоящий из одной или нескольких строк. В файле сценария могут размещаться несколько операторов комментария. Все их содержимое последовательно заносится в нижнее окно конфигуратора при загрузке сценария.

[Show]
Label :)) O'K check O'K :))
Mark O’K
Value :)) Check O’K

Маркерный оператор.
Служит для выделения и занесения в протокол некоторых точек прохождения сценария. Дополнительно выводит текущее значение номера CAN узла *node-ID*.
Поле метки оператора Label является опционным. Если метка присутствует, она должна быть уникальна для всего сценария. В поле метки оператора строчные и прописные буквы не различаются. Максимальная длина метки 31 символ.
Опционное поле Mark может содержать до трех символов и является отметкой завершения

оператора в протоколе сценария (см. Протокол выполнения сценария.)
Опционное поле Value типа VISIBLE_STRING может использоваться для размещения дополнительной информации (до 31 символа) в строке протокола.

Управляющие операторы.

Участвуют в формировании алгоритма сценария, но не взаимодействуют с CAN сетью и CANopen устройствами.

[ActiveNode]
Label Active node label
NodeId 21
Value True/False
Goto Goto if active/inactive
OnError :((Invalid node

Оператор выбора активных CANopen узлов.
Поле метки оператора Label является опционным. Если метка присутствует, она должна быть уникальна для всего сценария. В поле метки оператора строчные и прописные буквы не различаются. Максимальная длина метки 31 символ.
Поле номера CAN узла NodeId является опционным и может быть задано в пределах от 1 до 127. Если поле NodeId задано, то его значение записывается в глобальную переменную *node-ID* и используется при выполнении оператора. Если это поле опущено, используется текущее значение глобальной переменной *node-ID*.
Обязательное поле **Value** имеет логический тип. Если его значение True и узел *активен*, выполнятся переход на оператор с меткой, заданной в поле **Goto**. Если значение **Value** - False и узел *не* активен, также выполнятся переход на оператор с заданной в поле **Goto** меткой.
В операторе может быть определено поле перехода по ошибке OnError. Тогда при ее возникновении (для оператора **[ActiveNode]** это нулевой номер узла сети) выполняется переход на указанную в OnError метку.
При запуске сценария на выполнение состояние всех узлов сети устанавливается активным. Проверка узла на фактическую активность осуществляется оператором **[CheckNode]**.

[AnalyzerOn]

Оператор включения анализатора CANopen траффика. Результат анализа записывается в файл протокола выполнения сценария в формате, аналогичном используемому в модуле CANopen анализатора. Однако, первичные данные CAN кадра и его временн*а*я метка не заносятся в файл протокола.

[AnalyzerOff]

Оператор отключения анализатора CANopen траффика. Прекращает запись результатов анализа в файл протокола выполнения сценария.

[Delay]
Label Delay label
Value 12.3

Оператор временн*о*й задержки с разрешением 0.1 секунды.
Поле метки оператора Label является опционным. Если метка присутствует, она должна быть уникальна для всего сценария. В поле метки оператора строчные и прописные буквы не различаются. Максимальная длина метки 31 символ.

Поле величины задержки **Value** обязательно. Задержка задается в секундах вещественным числом большим либо равным 0.1 с одним знаком после запятой. Максимальная величина задержки ограничена.

[Globals]
Label Globals label
NodeId 77
// NodeId++
// NodeId—
SYNC_counter 10

Оператор задает значения глобальных переменных и параметров сценария.
Поле метки оператора Label является опционным. Если метка присутствует, она должна быть уникальна для всего сценария. В поле метки оператора строчные и прописные буквы не различаются. Максимальная длина метки 31 символ.
Поле номера CAN узла NodeId является опционным. Оно задает новое значение глобальной переменной *node-ID* и может быть записано тремя способами:

- Абсолютное значение номера узла в пределах от 0 до 127.
- Инкремент (увеличение на единицу) номера узла: NodeId++. Если текущее значение номера узла равно 127, инкремент не производится.
- Декремент (уменьшение на единицу) номера узла: NodeId—. Если текущее значение номера узла равно 0, декремент не производится.

Поле значения SYNC счетчика SYNC_counter является опционным. Оно задает новое значение глобальной переменной *SYNC-counter* и должно находиться в диапазоне от 1 до 240. При запуске сценария этой переменной присваивается значение 1. Значение *SYNC-counter* выводится в протокол выполнения сценария в колонке основного параметра оператора.

[Goto]
Label Goto label
Goto Some operator label

Оператор безусловного перехода.
Поле метки оператора Label является опционным. Если метка присутствует, она должна быть уникальна для всего сценария. В поле метки оператора строчные и прописные буквы не различаются. Максимальная длина метки 31 символ.
В обязательном поле **Goto** задается метка оператора, на который выполняется переход.

[LoopBegin]
Label Loop begin label
Value 11

Оператор начала цикла. Тело цикла ограничивается операторами **[LoopBegin]** и **[LoopEnd]**. Возможен любой уровень вложенности циклов.
Поле метки оператора Label является опционным. Если метка присутствует, она должна быть уникальна для всего сценария. В поле метки оператора строчные и прописные буквы не различаются. Максимальная длина метки 31 символ.
Обязательное поле **Value** задает число итераций цикла. Значения 0 и 1 эквивалентны.

[LoopEnd]
Label Loop end label

Оператор окончания цикла. Тело цикла ограничивается операторами **[LoopBegin]** и

[LoopEnd]. Возможен любой уровень вложенности циклов.
Поле метки оператора Label является опционным. Если метка присутствует, она должна быть уникальна для всего сценария. В поле метки оператора строчные и прописные буквы не различаются. Максимальная длина метки 31 символ.

[Stop]
Label Stop label
Mark **
Value Additional information

Оператор завершения сценария (останова). Используется для задания дополнительных точек останова сценария.
Поле метки оператора Label является опционным. Если метка присутствует, она должна быть уникальна для всего сценария. В поле метки оператора строчные и прописные буквы не различаются. Максимальная длина метки 31 символ.
Опционное поле Mark может содержать до трех символов и является отметкой завершения оператора в протоколе сценария (см. Протокол выполнения сценария.)
Опционное поле Value типа VISIBLE_STRING может использоваться для размещения дополнительной информации (до 31 символа) в строке протокола.

CANopen операторы.

Участвуют в формировании алгоритма сценария, осуществляя взаимодействие с CANopen устройствами в составе CAN сети.

[CheckNode]
Label Check node label
NodeId 21
Value 0x000F0191
OnError :((Check node error

Проверяет, является ли узел CANopen сети активным или нет и соответственно его отмечает. Для этого **[CheckNode]** осуществляет считывание объекта 1000h (тип устройства) для узла *node-ID*. Если возникает тайм-аут SDO протокола, узел помечается как *не* активный. В случае успешного считывания типа устройства отнесение узла к активному либо не активному зависит от наличия и значения поля Value. Если Value не определено, устройство будет отнесено к активным, т.е. отбираются все узлы сети, находящиеся в пред-операционном или операционном состояниях. Если поле Value присутствует, производится его сравнение со считанным значением типа устройства и лишь при их совпадении узел помечается как активный. В этом случае отбираются CANopen устройства определенного типа.
Поле метки оператора Label является опционным. Если метка присутствует, она должна быть уникальна для всего сценария. В поле метки оператора строчные и прописные буквы не различаются. Максимальная длина метки 31 символ.
Поле номера CAN узла NodeId является опционным и может быть задано в пределах от 1 до 127. Если поле NodeId задано, то его значение записывается в глобальную переменную *node-ID* и используется при выполнении оператора. Если это поле опущено, используется текущее значение глобальной переменной *node-ID*.
Опционное поле Value (тип данных UNSIGNED32) определяет искомое значение объекта 1000h - тип устройства.
В операторе может быть определено поле перехода по ошибке OnError. Тогда при ее возникновении (для оператора **[CheckNode]** это любая ошибка, *кроме* тайм-аута SDO

протокола) выполняется переход на указанную в OnError метку.

[NMT]
Label NMT label
NodeId 12
Command Reset_Node
OnError :((Invalid node

Реализует протокол сетевого менеджера NMT.
Поле метки оператора Label является опционным. Если метка присутствует, она должна быть уникальна для всего сценария. В поле метки оператора строчные и прописные буквы не различаются. Максимальная длина метки 31 символ.
Поле номера CAN узла NodeId является опционным и может быть задано в пределах от 0 до 127. Если поле NodeId задано, то его значение записывается в глобальную переменную *node-ID* и используется при выполнении оператора. Если это поле опущено, используется текущее значение глобальной переменной *node-ID*. Нулевой номер узла может использоваться только в операторе **[NMT]**, где означает широковещательную NMT команду.
В обязательном поле **Command** записывается строка NMT команды. Поддерживаются следующие команды:
- Start_Node
- Stop_Node
- Enter_Pre-Operational
- Reset_Node
- Reset_Communication

В операторе может быть определено поле перехода по ошибке OnError. Тогда при ее возникновении (не верный номер узла сети) выполняется переход на указанную в OnError метку. Однако, в случае оператора **[NMT]** ошибка номера узла возникать не должна.

[Sync]
Label Sync label

Осуществляет передачу в CAN сеть объекта синхронизации SYNC со значением CAN-ID 0x80.
Поле метки оператора Label является опционным. Если метка присутствует, она должна быть уникальна для всего сценария. В поле метки оператора строчные и прописные буквы не различаются. Максимальная длина метки 31 символ.
Оператор **[Sync]** обычно используется совместно с операторами цикла.

[Sync_1]
Label Sync 1 label

Осуществляет передачу в CAN сеть объекта синхронизации SYNC со значением CAN-ID 0x80 и длиной поля данных 1 байт (версии 4.2 стандарта DS301). Поле данных заполняется значением SYNC счетчика.
Поле метки оператора Label является опционным. Если метка присутствует, она должна быть уникальна для всего сценария. В поле метки оператора строчные и прописные буквы не различаются. Максимальная длина метки 31 символ.
В качестве значения SYNC счетчика используется текущая величина глобальной переменной *SYNC-counter*. После использования в SYNC объекте значение этой переменной автоматически инкрементируется. Когда значение *SYNC-counter* превышает 240, оно переустанавливается в 1. Значение SYNC счетчика выводится в протокол выполнения сценария в колонке основного параметра оператора.

Оператор **[Sync_1]** обычно используется совместно с операторами цикла, параметры которых (число итераций) задают максимальное значение SYNC счетчика.

[Object]
Label Object label
CobId NodeId + 0x200
Length 7
Value 0x50 0x44 0x4F 0x6E 0x6F 0x64 0x65
RTR False

Формирует и отправляет в CAN сеть объект канального уровня с 11 битовым идентификатором коммуникационного объекта.
Поле метки оператора Label является опционным. Если метка присутствует, она должна быть уникальна для всего сценария. В поле метки оператора строчные и прописные буквы не различаются. Максимальная длина метки 31 символ.
Обязательное поле **CobId** задает идентификатор CAN кадра канального уровня. Идентификатор может быть записан как в виде абсолютного значения, так и с помощью формулы вида NodeId + 123. В последнем случае итоговым значением идентификатора будет сумма глобальной переменной *node-ID* и указанного в формуле числа.
В обязательном поле **Length** указывается длина поля данных CAN кадра: от 0 до 8 байт.
В опционном поле Value побайтно задается значение объекта. Если число указанных значений байт меньше длины кадра, недостающие полагаются равными нулю. Если же указано свыше восьми значений, то девятое и все последующие игнорируются.
Опционное поле RTR определяет тип кадра: True для кадра удаленного запроса и False для кадра данных. Значение по умолчанию – False (кадр данных).

[Read]
Label Read label
NodeId 0x12
Index 0x1000
SubInd 0x0
DataType unsigned32
Value 0x00010000
Unequal Read data not equal
OnError :((Read error

Осуществляет чтение (upload) объекта из узла CANopen сети. При чтении используется ускоренный либо сегментированный SDO протокол.
Поле метки оператора Label является опционным. Если метка присутствует, она должна быть уникальна для всего сценария. В поле метки оператора строчные и прописные буквы не различаются. Максимальная длина метки 31 символ.
Поле номера CAN узла NodeId является опционным и может быть задано в пределах от 1 до 127. Если поле NodeId задано, то его значение записывается в глобальную переменную *node-ID* и используется при выполнении оператора. Если это поле опущено, используется текущее значение глобальной переменной *node-ID*.
Обязательные поля **Index** и **SubInd** задают соответственно индекс и субиндекс объектного словаря CANopen slave устройства. Объект считывается по адресу, определяемому этими двумя параметрами.
В обязательном поле **DataType** указывается тип данных объекта. Он определяется своим названием и записывается в виде строки, например unsigned32, строчные и прописные буквы не различаются. Названия типов данных приведены в пункте «Поддерживаемые типы данных.»

Опционные поля Value и Unequal могут использоваться только совместно. Если определены оба поля, то после успешного завершения транзакции чтения данных из CANopen устройства производится сравнение значения объекта из поля Value с его величиной, считанной из устройства. При неравенстве значений выполняется переход на метку, определенную в Unequal. В протоколе сценария ситуация неравенства отмечается одним символом * - информация к сведению, а строка протокола дополнительно выделяется голубым цветом.
В операторе может быть определено поле перехода по ошибке OnError. Тогда при ее возникновении выполняется переход на указанную в OnError метку.

[Write]

Label	Write label
NodeId	77
Index	0x2000
SubInd	0x0
DataType	VISIBLE_STRING
Value	Sample visible string
Length	32
OnError	:((Write error

Осуществляет запись (download) объекта в узел CANopen сети. Для записи используется ускоренный либо сегментированный SDO протокол.
Поле метки оператора Label является опционным. Если метка присутствует, она должна быть уникальна для всего сценария. В поле метки оператора строчные и прописные буквы не различаются. Максимальная длина метки 31 символ.
Поле номера CAN узла NodeId является опционным и может быть задано в пределах от 1 до 127. Если поле NodeId задано, то его значение записывается в глобальную переменную *node-ID* и используется при выполнении оператора. Если это поле опущено, используется текущее значение глобальной переменной *node-ID*.
Обязательные поля **Index** и **SubInd** задают соответственно индекс и субиндекс объектного словаря CANopen slave устройства. Запись объекта производится по адресу, определяемому этими двумя параметрами.
В обязательном поле **DataType** указывается тип данных объекта. Он определяется своим названием и записывается в виде строки, например unsigned32, строчные и прописные буквы не различаются. Названия типов данных приведены в пункте «Поддерживаемые типы данных.»
В обязательном поле **Value** задается значение передаваемого объекта. Интерпретация значения осуществляется соответственно типу данных, указанному в **DataType**.
Опционное поле Length может использоваться для указания точного числа передаваемых байт, если данные относятся к типу с переменной длиной (см. Поддерживаемые типы данных.). Диапазон значений Length - от 1 до 255. Если указанная длина превышает полное число байт данных, записанных в поле **Value**, то оставшимся данным присваивается нулевое значение. Если длина не указана, она определяется автоматически. Для типа данных VISIBLE_STRING это будет длина строки, заданной в поле **Value**.
В операторе может быть определено поле перехода по ошибке OnError. Тогда при ее возникновении выполняется переход на указанную в OnError метку.

Протокол выполнения сценария.

После запуска сценария протокол его выполнения заносится в верхнее окно конфигуратора и в текстовый файл. По умолчанию для файла протокола используется имя файла сценария с расширением slg. Возможен выбор произвольного имени файла протокола с использованием кнопки Select log вверху окна конфигуратора. Запись протокола в файл является обязательной.

Каждый исполненный оператор сценария отображается в протоколе отдельной строкой, которая содержит следующие колонки (поля):

- Статус или отметка завершения оператора (безымянная колонка). В случае нормального завершения остается пустой. Может содержать один символ * - информация к сведению, при этом строка протокола в окне конфигуратора дополнительно выделяется голубым цветом. Два символа ** означают предупреждение и дополнительно отмечают строку протокола светло-желтым цветом. А три символа *** указывают на серьезную ошибку - строка протокола будет выделена оранжевым цветом. Для операторов **[Show]** и **[Stop]** отметка (до трех символов) может быть задана в поле Mark. Если оно содержит символы *, то строка протокола получает дополнительное цветовое выделение, как описано выше. Если же поле Mark оператора **[Show]** не определено, либо не содержит символов *, соответствующая строка протокола в окне конфигуратора выделяется светло-зеленым цветом.
- **Step**. Номер шага сценария.
- **Operation**. Название исполняемого оператора сценария.
- **Label**. Метка оператора.
- **Node**. Номер CAN узла.
- **Index**. Индекс объектного словаря CAN устройства.
- **SubInd**. Суб-индекс объектного словаря CAN устройства.
- **DataType**. Название типа данных.
- **Value**. Значение основного параметра оператора. Для данных переменной длины в протокол выводятся до 31 первых символа.
- **ValComp**. Значение, с которым сравнивается основной параметр оператора. Для данных переменной длины в протокол выводятся до 31 первых символа.
- **Transaction**. Описание результата выполнения оператора (транзакции).
- **TimeStamp**. Временная метка начала выполнения оператора в формате DD-MM-YYYY HH:MM:SS

Поддерживаемые типы данных.

Данные фиксированной длины.

- 0001h - BOOLEAN.
- 0002h - INTEGER8; 0005h - UNSIGNED8.
- 0003h - INTEGER16; 0006h - UNSIGNED16.
- 0010h - INTEGER24; 0016h - UNSIGNED24.
- 0004h - INTEGER32; 0007h - UNSIGNED32.
- 0012h - INTEGER40; 0018h - UNSIGNED40.
- 0013h - INTEGER48; 0019h - UNSIGNED48.
- 0014h - INTEGER56; 001Ah - UNSIGNED56.
- 0015h - INTEGER64; 001Bh - UNSIGNED64.
- 0008h - REAL32;
- 0011h - REAL64.

Данные переменной длины.

Максимальная длина данных – 255 байт.

- 0009h - VISIBLE_STRING.