

## Блок переменных

Представим переменную как место хранения значения во встроенной памяти блока NXT. Другие программные блоки могут считывать текущее значение переменной (и даже менять его), связываясь с блоком через шины данных.

### Создание переменной

1. Чтобы создать переменную, выберите сначала команду Определить переменную в меню Правка программного приложения. В диалоговом окне задайте своей новой переменной короткое, легкое для понимания имя и укажите тип данных, который эта переменная будет поддерживать (т.е. текстовый, числовой или логический).
2. Затем перетащите блок переменных в свою программу и выберите имя переменной из списка.
3. Выберите режим "считывать" либо "записывать" переменную. Выбор режима записи переменной позволит вам в дальнейшем менять значение переменной, записывая ее новое значение поверх существующего. Чтобы это сделать, вам нужно будет подсоединить кодовую шину входных данных к концентратору данных блока, доставляющего новую информацию.
4. Вы можете задать постоянное исходное значение переменной, введя его с клавиатуры в окне ввода данных (или воспользовавшись селективными кнопками, если для вашей переменной указан логический тип данных).



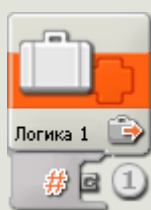
Примечание: если вам необходимо изменить тип данных или имя переменной, вернитесь к команде Определить переменную в меню Правка и внесите изменения в настройки.

### Использование переменной

Значение переменной со временем может меняться. Вы можете создать переменную под названием "Зарегистрированный уровень освещенности" и задать ей исходное значение 50. Затем, пока ваш робот (оборудованный световым датчиком) перемещается к определенному месту, вы можете использовать блок переменных (с настройкой "записывать") для регистрации уровня освещенности в этом месте. Чтобы это сделать, вы должны подключить кодовую шину выходных данных от блока датчика света к входному разъему блока переменных.

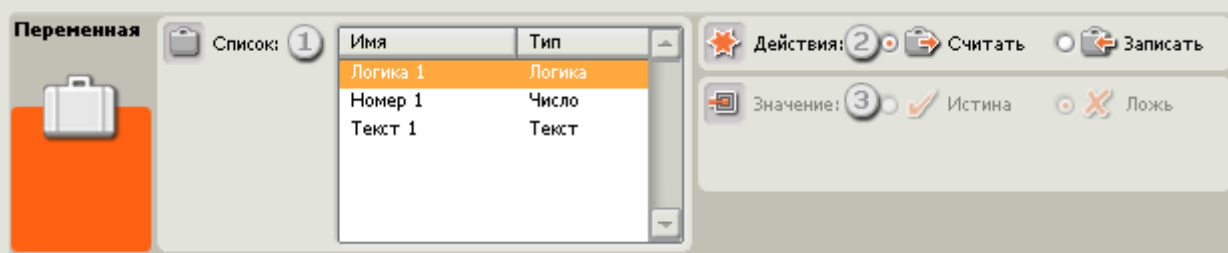
Впоследствии вы сможете использовать в своей программе блок переменных (с настройкой "считывать") для считывания значений уровня освещенности, зарегистрированных в этом месте. Другие блоки, подсоединенные к выходному разъему блока с помощью шины данных,

могут получать доступ к информации, сохраненной в переменной (т.е. зарегистрированному уровню освещенности).



1. После помещения программного блока в рабочую область концентратор данных блока откроется автоматически. Чтобы другие блоки могли считывать текущее значение переменной, к выходному разъему этого блока должна быть подсоединена шина данных. Чтобы записать новое значение переменной, вы должны также подсоединить шину входных данных к концентратору данных блока. (Чтобы получить дополнительную информацию, см. ниже раздел "Концентратор данных".)

### Настройка Программного блока переменных:



1. Выберите из списка переменную, которую вы создали, с помощью команды Определить переменную в меню Правка.
2. Выберите либо режим считывания текущего значения переменной, либо режим записи нового значения переменной.
3. Чтобы присвоить переменной постоянное значение, в окне ввода данных введите с клавиатуры число или текст, или воспользуйтесь селективными кнопками, если переменная относится к типу логических данных. Помните, что если к блоку подсоединена шина входных данных, а программный блок переменных установлен на режим "считывать", новая информация, динамически поступающая через шину данных, будет обладать преимуществом перед любым постоянным значением.



Примечание: если вам необходимо изменить тип данных или имя переменной, вернитесь к команде Определить переменную в меню Правка и внесите изменения в настройки.

### Совместное использование данных программой и Мой Блок

Чтобы сделать возможным совместное использование данных вашей основной программой и Мой Блок (т.е. двумя файлами .gbt), выполните две вещи:

1. Определите переменную с одним и тем же именем и типом данных в основной программе и в Мой Блок.
2. Используйте эту переменную в блоке переменных в основной программе и в Мой Блок.

Любое место, где используется эта переменная, будет выполнять считывание/запись в одних и тех же ячейках памяти NXT.

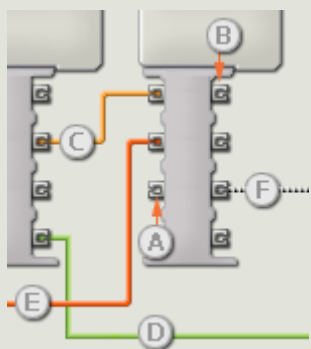
## Настройка концентратора данных программного блока переменных

Вы можете управлять блоком переменных динамически, подключая шины данных (от концентраторов данных других блоков) к концентратору данных блока переменных.

Поместив блок в рабочую область, откройте концентратор данных блока, щелкнув по кнопке в нижней левой части блока.



Шины данных, передающие входящие данные на блок, подключены к разъемам, находящимся на левой стороне их концентраторов данных. Шины данных, передающие исходящие данные, подключены к разъемам, находящимся на правой стороне.



- [A] Входной разъем
- [B] Выходной разъем
- [C] Числовая шина данных (желтая)
- [D] Логическая шина данных (зеленая)
- [E] Текстовая шина данных (оранжевая)
- [F] Поврежденная шина данных (серая)

## **Передача данных от входного разъема на выходной разъем**

Если для входного разъема есть соответствующий выходной разъем (см. выше А), входящие данные будут проходить от входного разъема на выходной разъем без изменений. В этом случае вы сможете использовать выходной разъем, если входной разъем подключен к шине данных входа; подключение шины данных выхода к такому выходному разъему без подключенной шины данных входа приведет к «неисправности» шины данных выхода (она будет окрашена в серый цвет).

## **Шины данных осуществляют передачу специфических типов данных**

Каждая шина данных осуществляет обмен специфическими типами данных между блоками. Например, если шина данных перетянута с логического разъема на концентратор данным блока, ее можно подключить только к логическому разъему на концентраторе данных другого блока. На приведенной ниже схеме показано, какой вид данных может принимать или передавать каждый из разъемов.

## **Окраски шин данных**

Шины данных различаются особыми окрасками: шины, передающие числовые данные, имеют желтый цвет; шины, передающие логические данные, имеют зеленый цвет; шины, передающие текстовые данные, имеют оранжевый цвет.

## **«Поврежденные» шины данных**

Если вы пытаетесь подключить шину данных к разъему несоответствующего типа данных, шина будет повреждена (ее цвет изменится на серый). Вы не сможете загрузить вашу программу, если шина данных повреждена.

Если вы щелкнете по поврежденной шине, то в небольшом окне контекстной справки в правом нижнем углу рабочей области вы сможете прочесть информацию о причине повреждения.


## **Данные должны соответствовать возможному диапазону, в котором работает разъем**

Если шина данных входа передает значение, не попадающее в допустимый диапазон разъема, к которому она подключена, блок либо проигнорирует такие данные, либо трансформирует данные, попадающие в этот диапазон. Для разъемов, допускающих небольшое количество значений (например, только 0, 1 или 2), разъем проигнорирует входящий сигнал, если передаваемое значение не будет вписываться в диапазон разъема.

Для разъемов, допускающих более широкий диапазон значений (например, 0 – 100), разъем усилит любой входящий сигнал, если он не соответствует диапазону. Например, если

разъем питания программного блока перемещения получает входящий сигнал со значением 150, блок преобразует входящее значение в 100 (т.е., в числовое значение, лежащее в пределах диапазона разъема мощности).

**На этой схеме показаны различные характеристики разъемов на концентраторе данных блока переменных:**

	<b>Разъем</b>	<b>Тип данных</b>	<b>Возможный диапазон</b>	<b>Что обозначают эти величины</b>	<b>Этот разъем игнорируется, если...</b>
	Значение*	Любое		Значение для считывания или для записи	

\*Этот разъем для данных меняет свой вид в зависимости от типа данных.