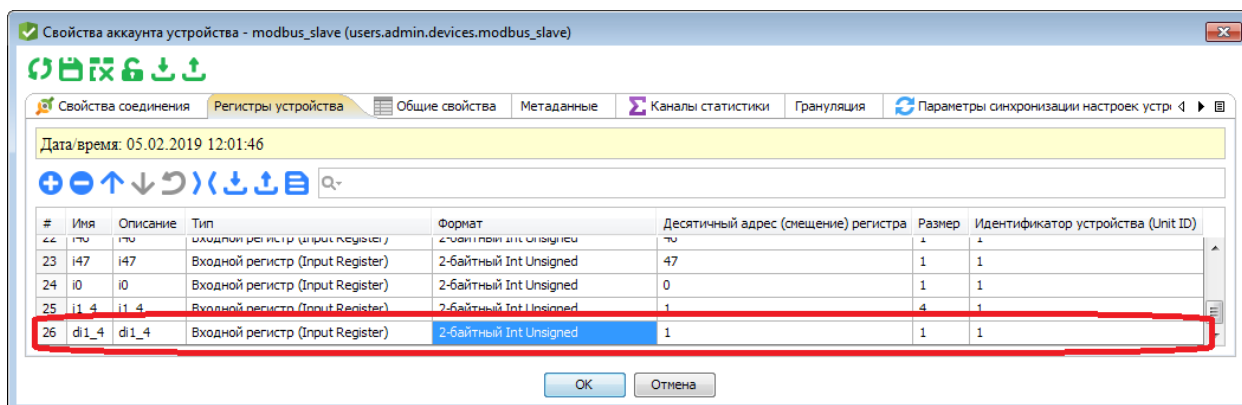


LANTANSoft. Распаковка дискретных сигналов

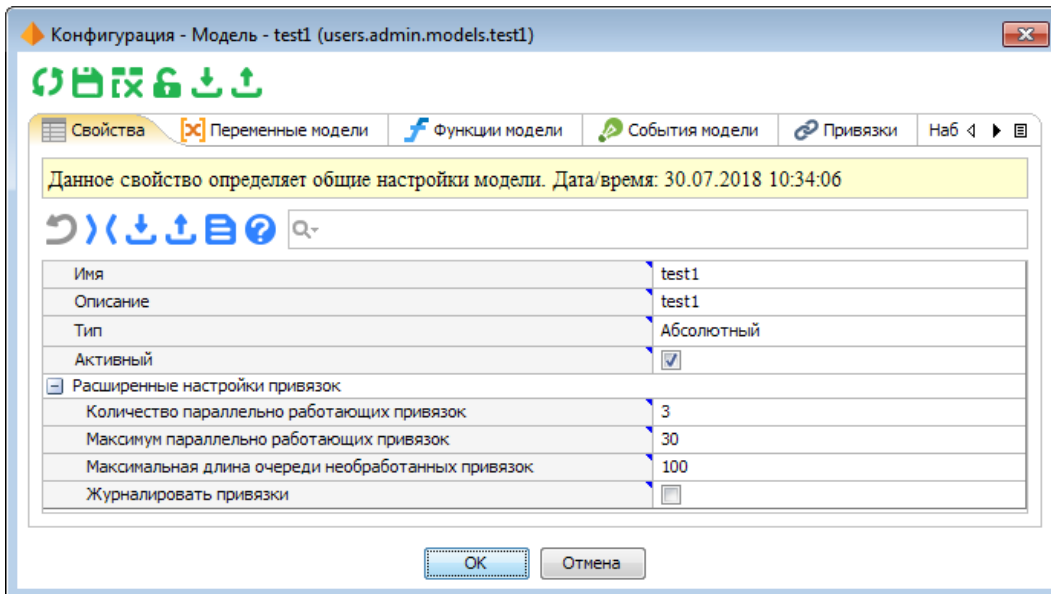
Если среда программирования контроллера поддерживает возможность запаковки дискретных сигналов в числа целого формата и передачи их по протоколу связи, то на стороне LantanSoft можно распаковывать такие сигналы в обратном порядке.

Данный способ имеет как достоинства, так и недостатки. К достоинствам можно отнести экономию тегов лицензии LantanSoft. Недостатком является некоторая временная задержка при получении результирующих сигналов (время запаковки дискретных сигналов в контроллере + время передачи запакованного числа + время распаковки сигналов в LantanSoft). Исходя из этого данный метод не рекомендуется применять для систем, в которых нужно свести к минимуму задержку между формированием сигнала и его регистрацией в LantanSoft.

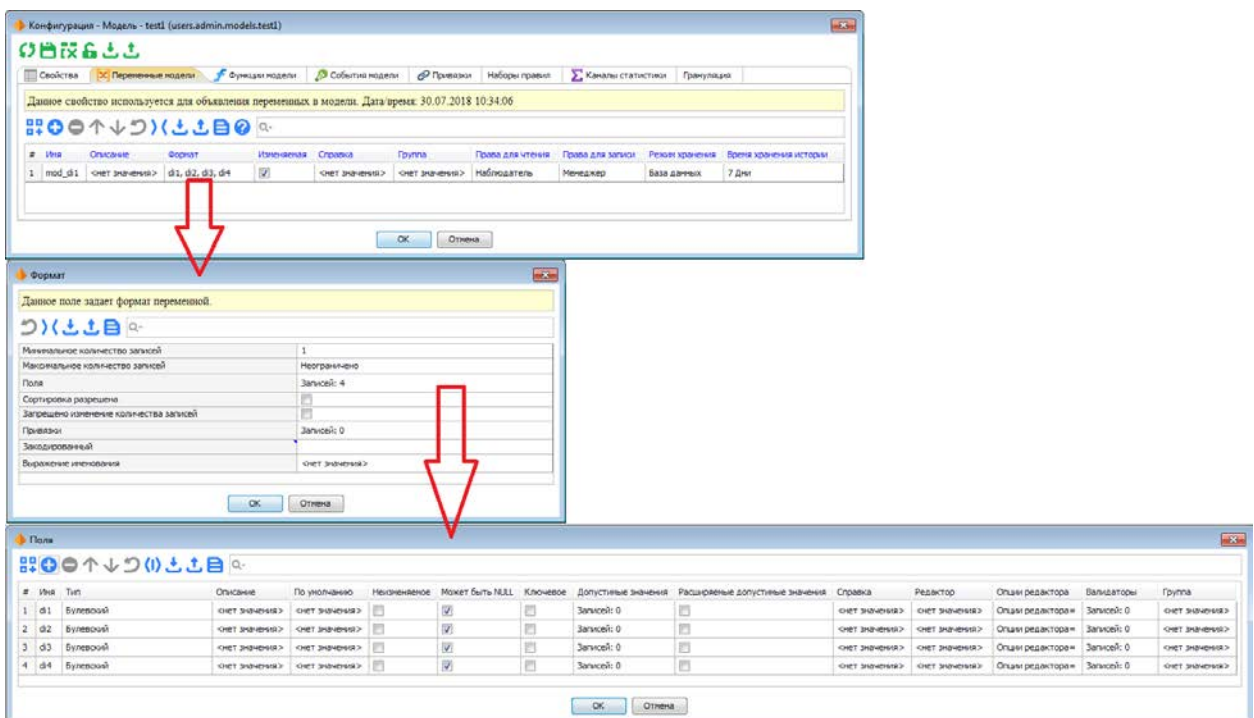
1. Предположим мы получаем из контроллера по протоколу Modbus целое число, в котором запакованы дискретные сигналы:



2. Чтобы извлечь отдельные биты из числа создадим модель, в которую добавим результирующие сигналы (в примере **test1**):

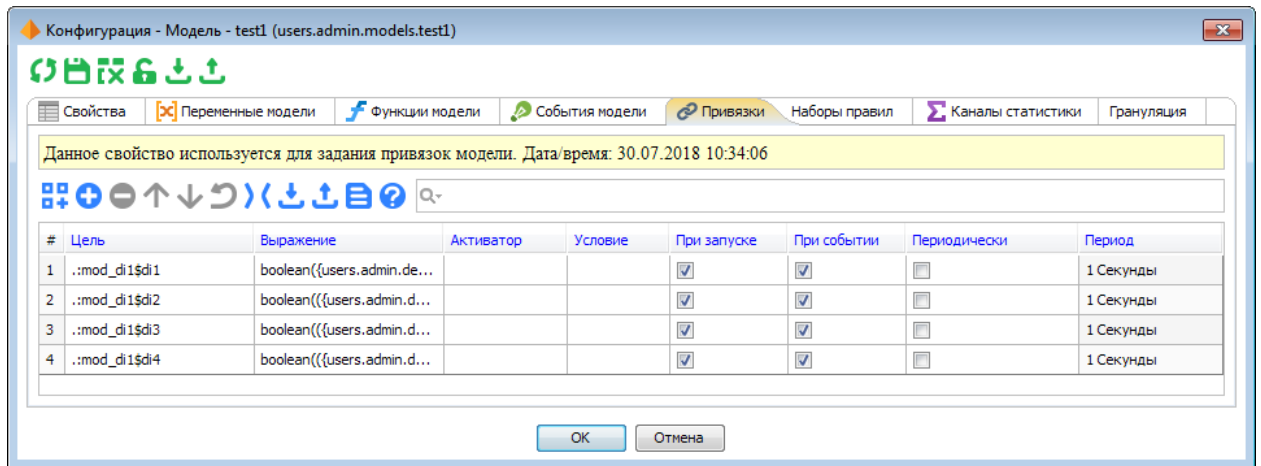


3. Предположим в исходном целом числе запакованы 4 дискретных сигнала. В модель добавим одну переменную (в примере `mod_di1`), в которую добавим 4 поля дискретного типа (в примере `di1`, `di2`, `di3`, `di4`):

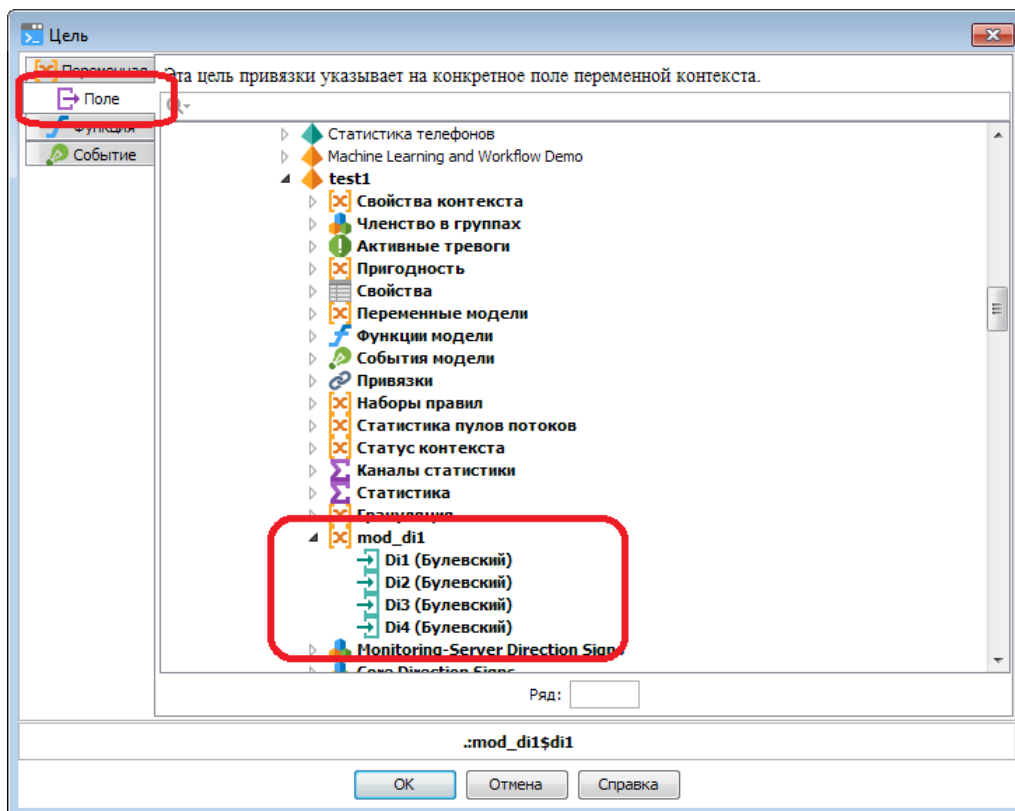


Примечание: При задании формата переменной необходимо выставить **Минимальное количество записей** в 1 (по умолчанию стоит 0).

В привязках модели необходимо добавить 4 привязки, целью которых будут поля переменной, добавленной выше:



Примечание: По умолчанию в окне цели выбрана вкладка **Переменная**, чтобы найти поля переменной необходимо выбрать вкладку **Поле**:



В поле **Выражение** для каждой привязки будем вставлять выражения «выдергивания» нужного бита из исходного числа. Общая формула получения нужного бита из числа следующая:

$boolean(\{a\} \gg n \& 1)$, где:

- **boolean ()** – функция приведения результата к дискретному типу;

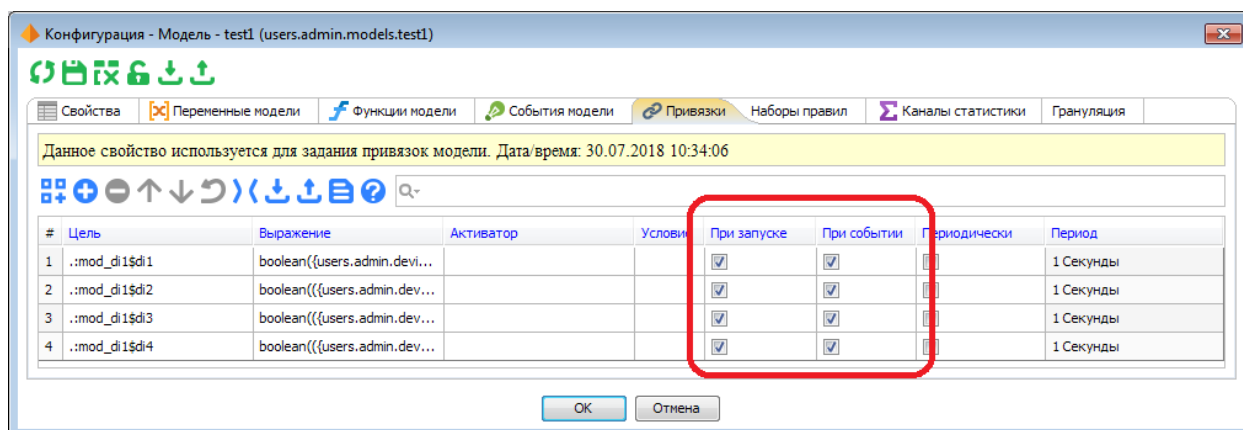
- {a} – абсолютный путь до исходного целочисленного сигнала;
- >> – оператор побитового смещения вправо;
- n – количество бит на которое необходимо сместить число;
- & - оператор побитового И.

Т.е. мы смещаем вправо число до необходимого бита и побитово умножаем результат на 1, чтобы получить значение первого (крайнего правого) бита.

Для нашего примера в полях **Выражение** будут следующие значения:

Цель	Выражение
.:mod_di1\$di1	boolean({users.admin.devices.modbus_slave:di1_4\$di1_4} & 1)
.:mod_di1\$di2	boolean(((users.admin.devices.modbus_slave:di1_4\$di1_4} >> 1) & 1)
.:mod_di1\$di3	boolean(((users.admin.devices.modbus_slave:di1_4\$di1_4} >> 2) & 1)
.:mod_di1\$di4	boolean(((users.admin.devices.modbus_slave:di1_4\$di1_4} >> 3) & 1)

Так же необходимо убедиться, что в Привязках выставлены галочки в полях **При запуске** и **При событии**. Это будет означать, что пересчет будет производиться при запуске сервера и при любом изменении исходного сигнала:

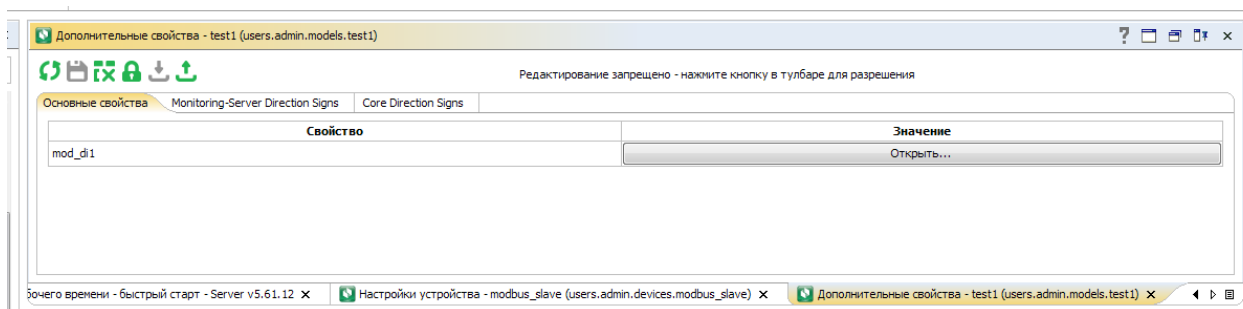


4. Жмем ОК, модель готова.

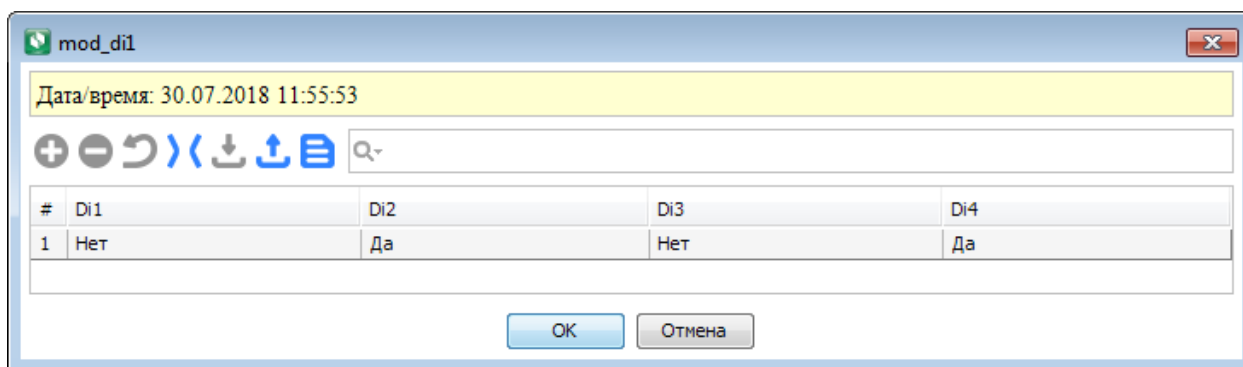
Результат работы модели можно посмотреть следующим образом:

Щелкаем по модели правой кнопкой мыши и в контекстном меню выбираем пункт **Редактировать дополнительные свойства**.

Откроется панель со списком переменных модели (в нашем примере 1 переменная **mod_di1**):



Чтобы посмотреть значения полей переменной в поле **Значение** жмем кнопку **Открыть**. При этом откроется табличка с значениями полей переменной:



Это и есть распакованные значения исходных дискретных сигналов.