

Контроллер для больших и маленьких энергообъектов

Откуда только ни приезжали в нашу редакцию интеллектуальные устройства: из Екатеринбурга, Германии, Швейцарии, Москвы. Но в этот раз гость пропитан интеллигентностью Северной столицы нашей необъятной страны.

На самом деле в Питере находится много производств, даже небольшую викторину можно было бы сделать, но рубрика у нас сугубо техническая, а не развлекательная. Представляем вам нового участника рубрики «ТестДрайв» — контроллер присоединения NPT ВАУ от компании «ЭнергопромАвтоматизация».

Не хватает смокинга

NPT ВАУ – самое серьезное из устройств NPT: крейт 6U и слоты под 14 плат. С первого взгляда на устройство хотелось спросить себя: «А что оно не умеет делать?». На рис. 1 представлена лицевая панель с большим ярким экраном, кнопками управления и ключом выбора режима управления (местное / дистанционное / выведено).



Рисунок 1. Лицевая панель NPT ВАУ

Много кнопок, много светодиодов. В базовом режиме на экран выводится мнемосхема контролируемого присоединения. Экран имеет высокое разрешение и хорошую читаемость. За лицевую панель сразу ставим пять с плюсом. Стоит отметить, что на наш вопрос о необходимости разработки устройства таких габаритов специалисты компании сразу ответили, что размеры крейтов у них представлены в 4 вариантах. Участник тест-драйва — это контроллер присоединения для подстанций 330–500 кВ. А вот платы во всех устройствах NPT одинаковые для удобства эксплуатации. Задняя часть контроллера NPT BAY содержит слоты с платами (рис.2).

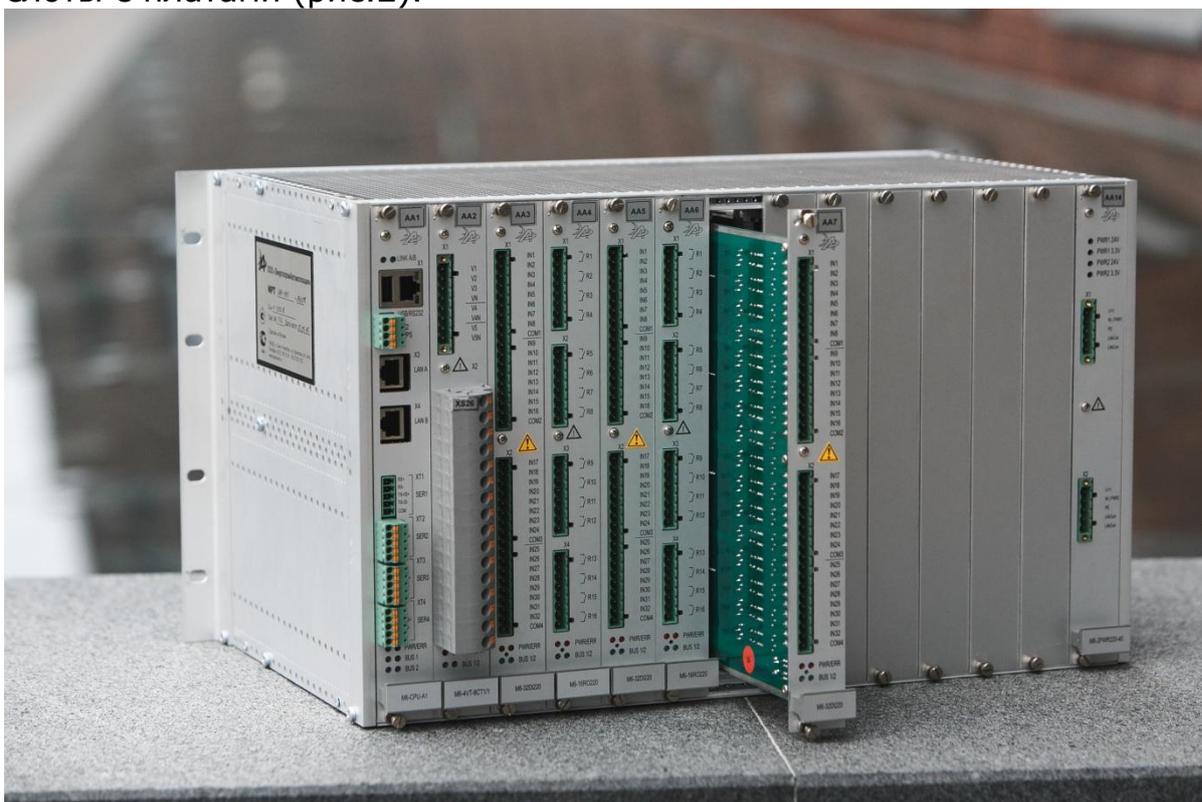


Рис. 2. Вид сзади

Как видно из рисунка, к нам приехал терминал с неслабой конфигурацией, но, к сожалению, среди такого изобилия плат мы не нашли платы приема SV. Поэтому об удобстве работы с потоком можем только гадать. Для проведения тест-драйва нам были предоставлены следующие платы: плата центрального процессора, плата измерений (подключение 2ТТ и 2ТН), 2 платы дискретных входов по 32 шт. на плату, две платы релейных выходов по 16 выходов на плату и плата питания с универсальными и резервированными входами питания. Плата центрального процессора оснащена 1 портом под RS 232, 4 портами с интерфейсом RS-485, дублированными портами Ethernet. Про возможности устройства с точки зрения участия в построении топологии ЛВС мы поговорим позже. Конечно, мы достали одну плату и оценили качество сборки. Предлагаем и вам оценить его (рис. 3).

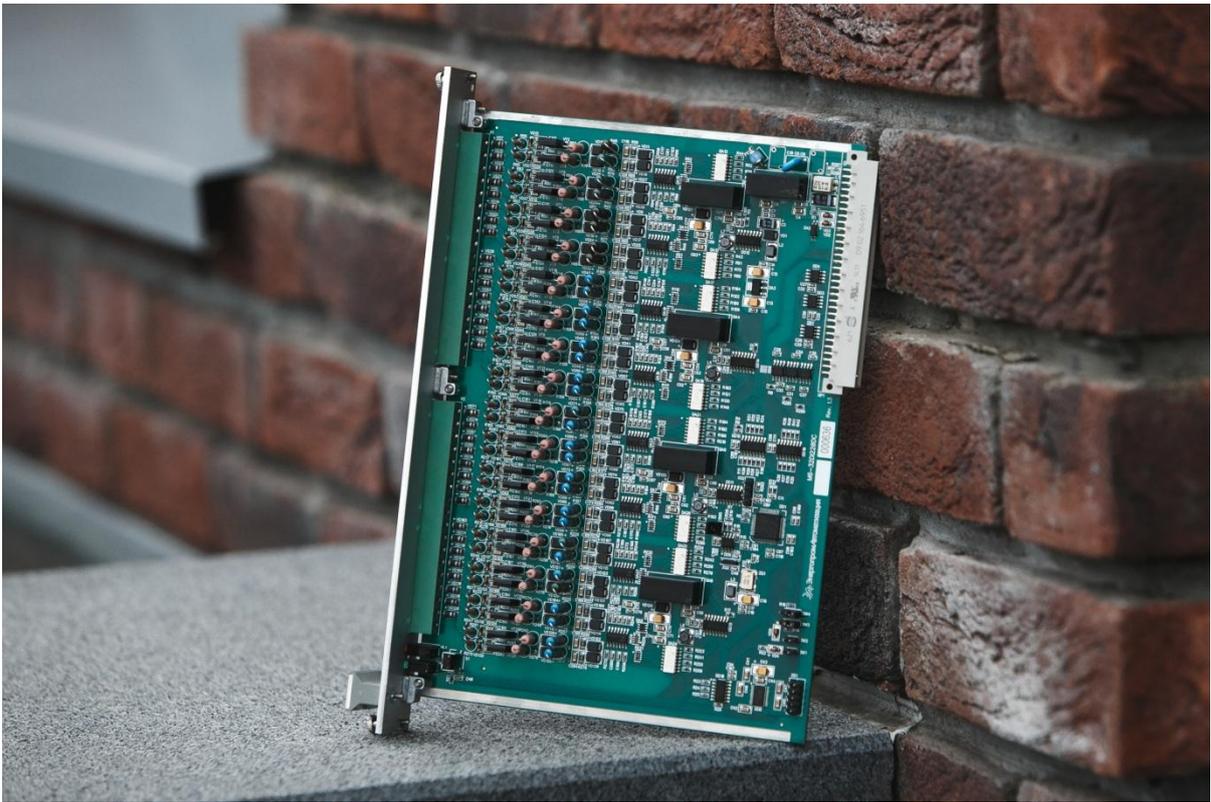


Рис. 3. Плата дискретных входов

Говоря о форм-факторе устройства, стоит отметить, что монтировать его можно как в шкаф, так и на монтажную панель, для этого предусмотрен вариант без лицевой панели. По количеству слотов также есть варианты, а с учетом большого количества разнообразных плат заказчик сможет набрать свой, индивидуальный терминал.

Встречай по одежке...

Первым впечатлением мы поделились. Но не столь важно, как выглядит устройство и насколько богатый у него ИЧМ. Гораздо важнее то, что оно в действительности может и как с ним совладать. Поигравшись с меню на ИЧМ, мы открыли конфигуратор. Первое, что пришлось сделать, — обновить Java. Ну куда без этого, как говорится. После этого перед нами предстал достаточно приятный интерфейс (рис. 4).

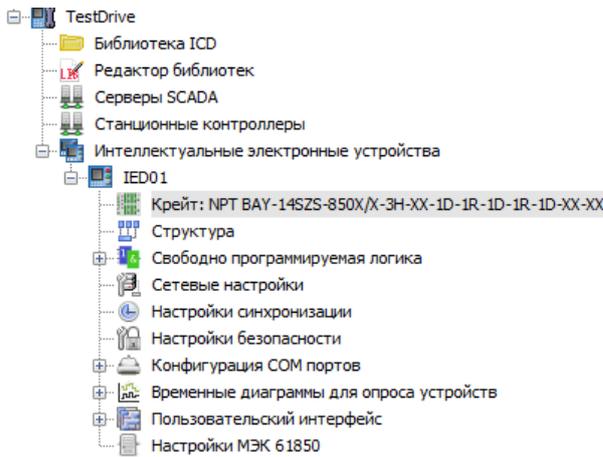


Рис. 6. Древо проекта

Так как в рамках рубрики «Тест-Драйв» мы стараемся максимально оценить легкость работы с устройством (ведь если работаешь с интуитивно понятным устройством, то и вероятность ошибки значительно меньше), то мы приступили к анализу основных интересующих нас пунктов без предварительного изучения мануала.

Сетевые настройки

NPT BAY — первое устройство, которое порадовало нас богатым разнообразием режимов работы портов в рамках одной платы центрального процессора. Раздел сетевых настроек представлен на рис. 7.

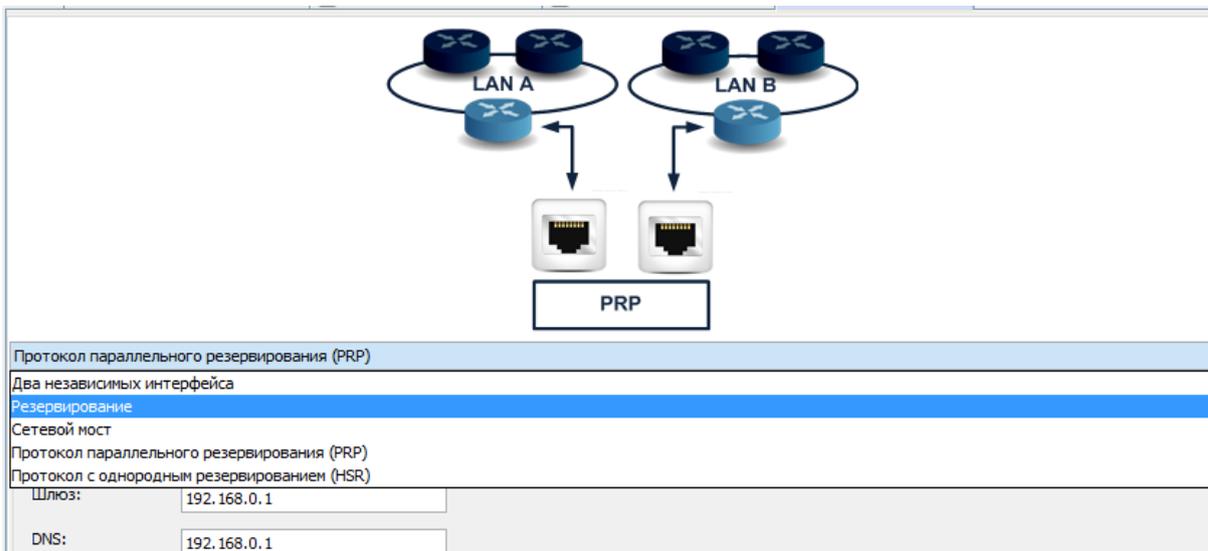


Рис. 7. Настройка сети

Как видно из рис. 7, устройство может работать в режиме горячего резервирования, сетевого моста, PRP и HSR. Пожалуй, это первое устройство с поддержкой HSR, которое пришло в нашу редакцию. В

зависимости от выбранного режима работы портов предлагаются разные варианты их настройки. Все удобно и понятно.

Назвался контроллером – контролируй!

Так как с самого начала мы поняли, что посмотреть на SV не получится, мы решили поработать со вторым многоадресным протоколом, описанным в IEC 61850 — GOOSE. Поскольку контроллер в общем своем назначении служит точкой сбора данных от устройств присоединения, то мы решили подписать устройство на GOOSEсообщение от терминала РЗА. И вот тут начали проявляться интересные особенности. В связи с тем, что устройство к нам пришло сконфигурированным и в WireShark мы обнаружили исходящее GOOSEсообщение, возник вопрос, как посмотреть текущую конфигурацию устройства. Оказалось, что выгрузить в конфигуратор текущую конфигурацию устройства просто невозможно. Производитель связывает отсутствие данной возможности с тем, что работа всегда ведется в проекте, где добавлены все устройства, а если бы была возможность выгрузить конфигурацию отдельно после запуска подстанции, то при ее обновлении могли бы быть допущены ошибки. С этим можно согласиться, но в отношении GOOSEсообщений данный подход мы считаем некорректным, особенно если в рамках одного энергообъекта будет множество устройств от разных вендоров. Для организации коммуникации по GOOSE требуется добавить в проект устройства приемники/источники данных сообщений. Такое устройство подтягивается посредством импорта CIDфайла. После добавления устройства мы перешли во вкладку «Структура» (рис. 8).

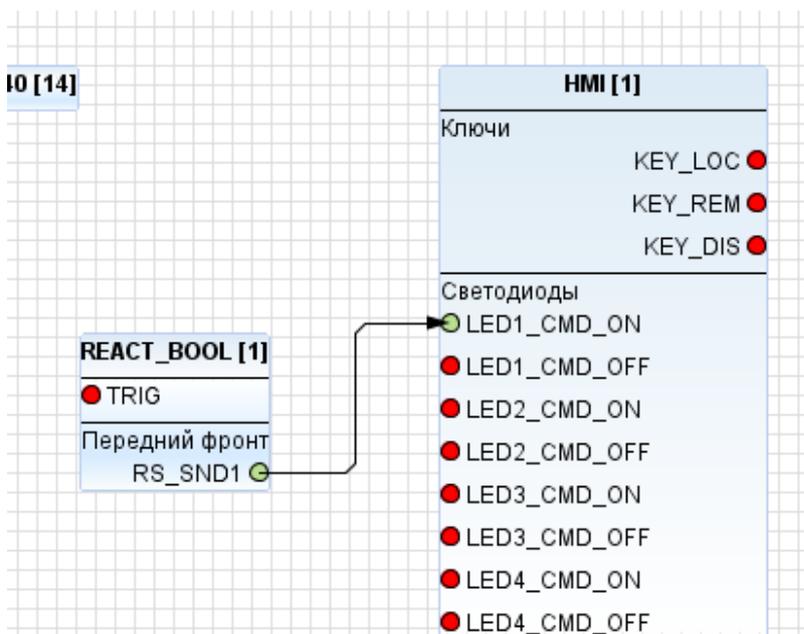


Рис. 8. Структура

Стоит отметить, что это основная вкладка для работы с устройством. На самой структуре после описания слотов устройства отображаются функциональные блоки. Заходя в настройки блока, можно отобразить или скрыть те входы и выходы, которые требуются. Мы, например, хотели, чтобы при получении GOOSE-сообщений загорался светодиод. Соответственно мы развернули функциональный блок HMI. Так как GOOSE-сообщение — это дискретный сигнал в цифровом виде, то для терминала требуется определить его, поэтому был введен блок REACT_BOOL. В настройках блока мы выбрали, что логическая единица во внутренней логике терминала, которая будет зажигать светодиод, будет формироваться по переднему фронту, соответственно осталось подвязать к блоку GOOSE-сообщение. Однако с первого раза у нас это не получилось. Разобраться в проблеме нам помог встроенный в конфигуратор функционал: возможность смотреть CID-файл в разметке SCL, как например в NotePAD++. В результате при открытии CID-файла в NotePAD и встроенном редакторе мы нашли очевидную разницу (рис. 9 и 10).

```

<Communication>
  <SubNetwork name="tekvel">
    <ConnectedAP apName="AP1" iedName="Sepam">
      <Address>
        <P type="IP" xsi:type="tP_IP">192.168.1.107</P>
        <P type="IP-SUBNET" xsi:type="tP_IP-SUBNET">255.255.255.0</P>
        <P type="IP-GATEWAY" xsi:type="tP_IP-GATEWAY">0.0.0.0</P>
        <P type="OSI-PSEL" xsi:type="tP_OSI-PSEL">00000001</P>
        <P type="OSI-SSEL" xsi:type="tP_OSI-SSEL">0001</P>
        <P type="OSI-TSEL" xsi:type="tP_OSI-TSEL">0001</P>
      </Address>
      <GSE cbName="Button" ldInst="LDO">
        <Address>
          <P type="MAC-Address" xsi:type="tP_MAC-Address">01-0C-CD-01-00-12</P>
          <P type="APPID" xsi:type="tP_APPID">0012</P>
          <P type="VLAN-ID" xsi:type="tP_VLAN-ID">000</P>
          <P type="VLAN-PRIORITY" xsi:type="tP_VLAN-PRIORITY">4</P>
        </Address>
        <MinTime multiplier="m" unit="s">8</MinTime>
        <MaxTime multiplier="m" unit="s">5000</MaxTime>
      </GSE>
    </ConnectedAP>
  </SubNetwork>
</Communication>

```

Рис. 9. CID-файл в NotePAD

```

<Communication>
  <SubNetwork name="Default" desc="Default communication network">
    <ConnectedAP iedName="Sepam" apName="AP1" desc="AP1/Sepam"/>
  </SubNetwork>
</Communication>

```

Рис. 10. CID-файл в конфигураторе

Ошибка — при загрузке CID в конфигуратор не подтягивался раздел «Communication». Дополнительно мы протестировали встроенную функцию проверки SCL-файлов — сравнили ее с iec61850Bot —

бесплатным сервисом по работе с SCL в мессенджере Telegram. В нем, кстати, во время проведения тестдрайва члены сообщества «Цифровая подстанция» отдельно от редакции начали обсуждать NPT BAY. Разработчик в течение часа исправил проблему и появилась возможность на вход блока подать сигнал из GOOSEсообщения (рис. 11).

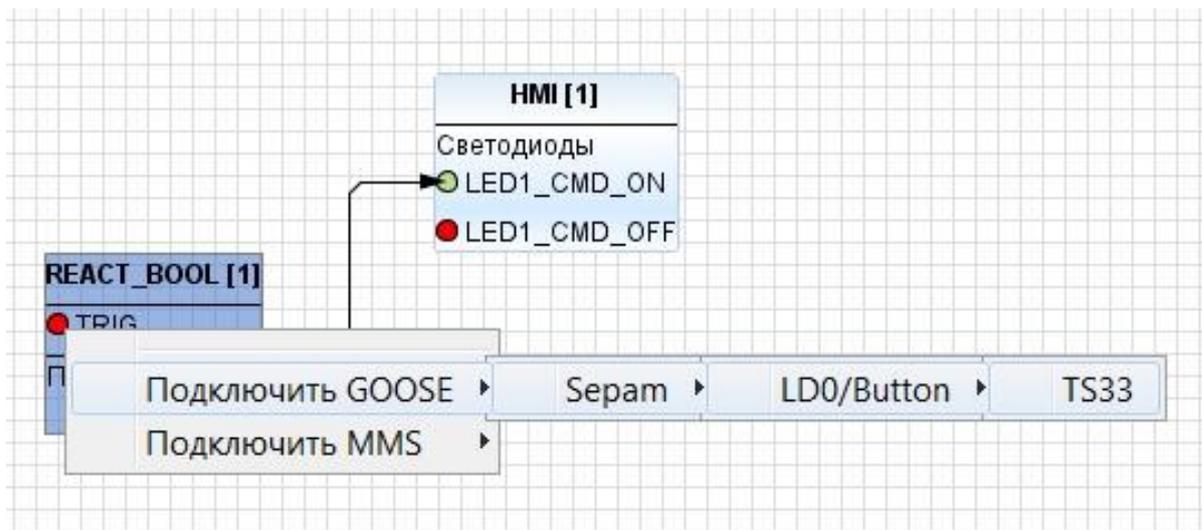


Рис. 11. Подключение GOOSE

Важно отметить, что конфигуратор сам определяет набор данных и блоки управления GOOSE-сообщениями, предоставляя возможность наладчику выбрать требуемый.

Гибкая логика

Не менее интересна работа с гибкой логикой. Пользователю достаточно создать рабочее поле, входные и выходные сигналы, внутренние сигналы, если потребуется, и, используя базовые функции, собрать свой алгоритм (рис. 12, 13, 14).

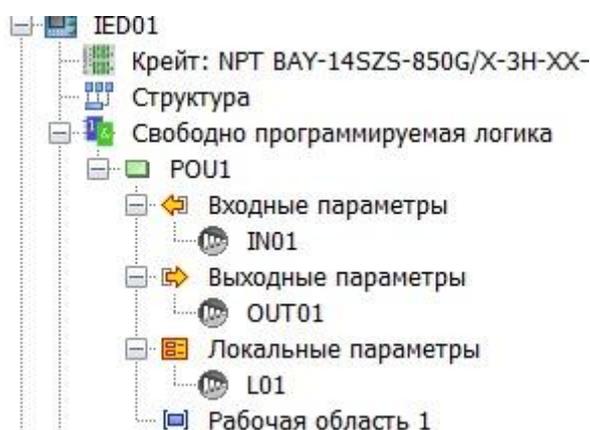


Рис. 12. Создание логического алгоритма

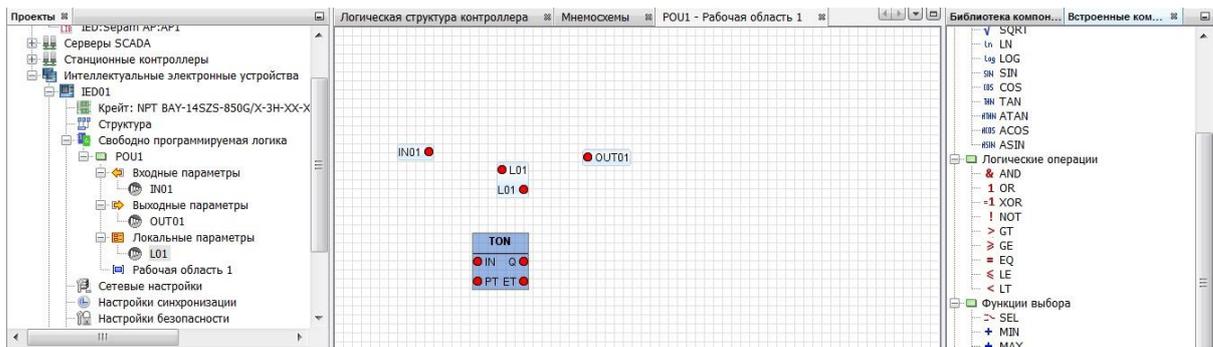


Рис. 13. Поле СПЛ

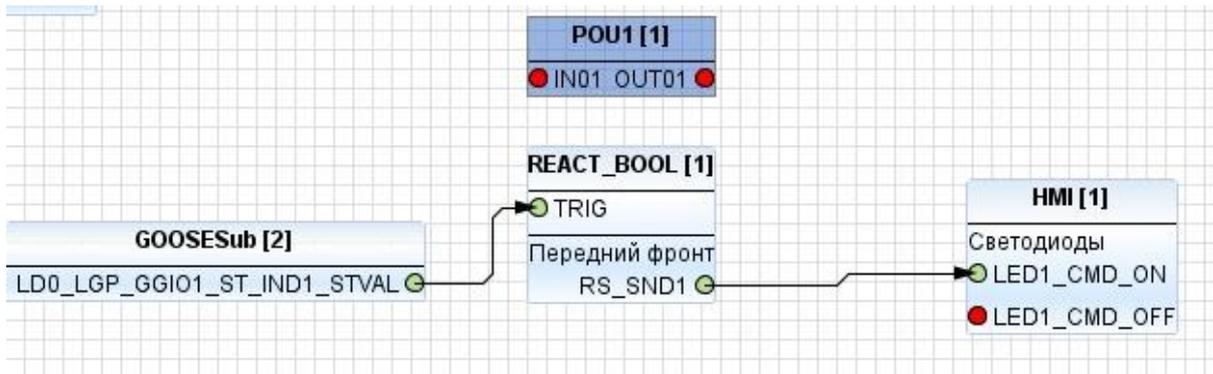


Рис. 14. Отображение алгоритма СПЛ в структуре

«Устройство заточено под функции контроллера присоединения», так производитель утверждает, что для решения типовых задач управления присоединением не требуется самостоятельно создавать базовые логические модули — достаточно использовать предустановленные блоки, подключив к ним сигналы и команды управления, для реализации логики в рамках контроллера присоединения».

Выводы

NPT BAY — это большой многофункциональный контроллер, который очень дружелюбен к пользователю благодаря продуманному и проработанному интерфейсу конфигуратора. Думаем, он придется по вкусу многим заказчикам. Хочется еще раз отметить его текущие ограничение— отсутствие возможности считывать и анализировать текущую конфигурацию. Возможно, мы просто привыкли к этому, пользуясь устройствами других вендоров, однако хотелось бы увидеть данную возможность в следующем релизе конфигуратора.