

```

Class Device
  ' Статические данные
  {“.” UInt16 { Min 300 Max 320}}
  'Конструктор
  || | Timer (180000,Value) #При запуске устанавливаем таймер и подписку на метод»
  || : 'Инициализация
  'Атрибуты Value – значение датчика и два счетчика таймера.
  { UInt16 Value #Значение датчика с подписками на выполнение и изменение значения .»
  || | = “I” #Адресация для выполнения. Присваиваем значение с интерфейса датчика.»
  :: ~ (EMin, Emax) #По событию изменение значения проверяем на условия.»
  UInt32 TimerD 180000 #Счетчик таймера датчика и начальное значение. 3 минуты.»
  UInt32 TimerP 4000 #Счетчик таймера насоса и начальное значение»
  :: #Подписка на срабатывание таймера. Значение 0»
  { “PA04”=0 “PA05”=0} #Выключение обоих насосов по истечению цикла.»
  'События EMin и EMax с подписками для включения соответствующих насосов
  ' с запуском таймера
  ~ EMin Value, Min < { “Pa04” = 1 Timer.Start(4000, TimerP)
  ~ EMax Value, Max > { “Pa05” = 1 Timer.Start(4000,TimerP)
}

```

Статические данные просто продемонстрировал.

Дальше идет конструктор. Т.е. метод запускаемый при создании объекта. (Событие «выполнение» В конструкторе создаем и запускаем таймер для чтения каждые 3 минуты. Второй параметр- адрес, к которому адресуется таймер по завершению счета. Т.е. каждые три минуты происходит адресация к атрибуту Value в котором подписки да два события «Выполнение» и изменение значения. По событию выполнению (т.е. по таймеры читаем новое значение с интерфейса (тут надо уточнить), а по изменению значения адресуемся к двум событиям EMin и EMax где происходит проверка на указанные диапазоны. При нарушении диапазона контроля запускаем соответствующие насосы, с контролем по таймеру для их выключения.