

## Установка бита в многоразрядном теге

Зачастую необходимо контролировать или управлять состоянием сразу нескольких одинаковых объектов. Для этого создаётся многоразрядный тег в одном из форматов – 8, 16 или 32-разрядном без знака. Каждый бит этого тега «отвечает» за состояние одного объекта. В примере создан 8-разрядный внутренний тег Imit5.

Например, есть два двигателя, состояние которых задаётся четырьмя кнопками:

- 1) Мотор 1 работает – бит 0 установлен;
- 2) Мотор 1 стоит – бит 0 сброшен;
- 3) Мотор 2 работает – бит 1 установлен;
- 4) Мотор 2 стоит – бит 1 сброшен;

Создаём кнопки и связываем их с выбранными битами тега с помощью Мастера динамики Setting/Resetting a bit (рис. 1). Данный мастер позволяет выбрать действие мыши, по которому будет производиться действие (рис. 2), что именно (установка или сброс бита) – рис. 3 и какой именно бит в многоразрядном теге изменяется (рис. 4).

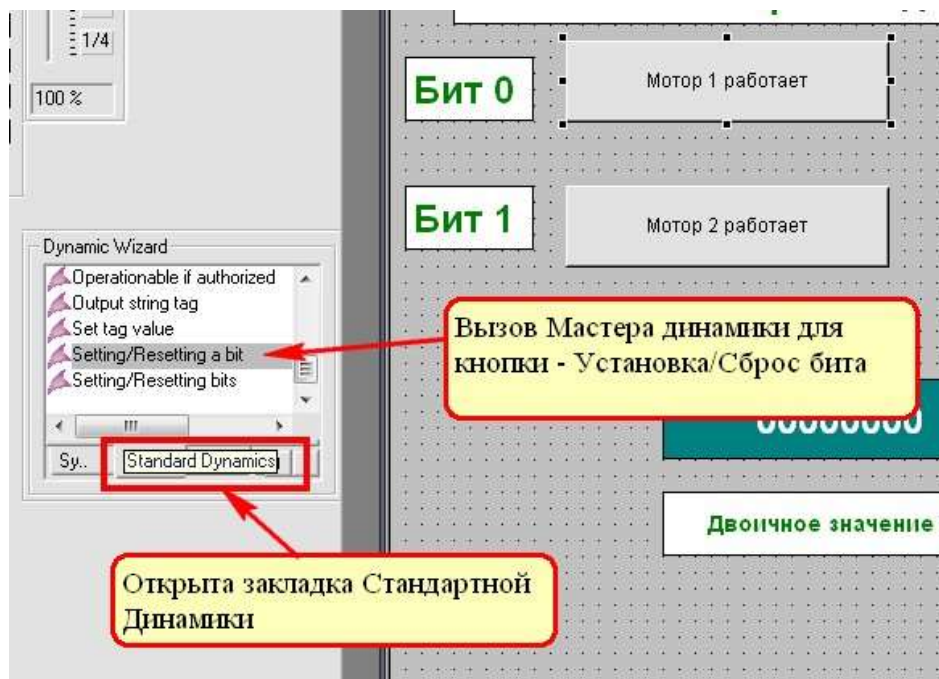


Рис. 1. Выбор мастера установки или сброса бита

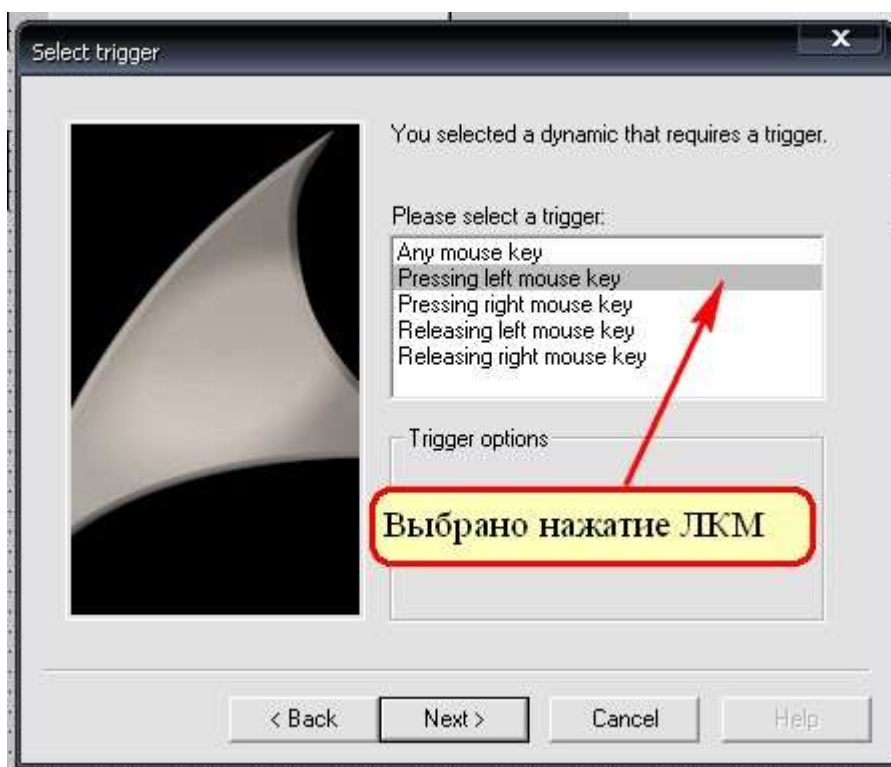


Рис. 2. Выбор действия мыши для активизации

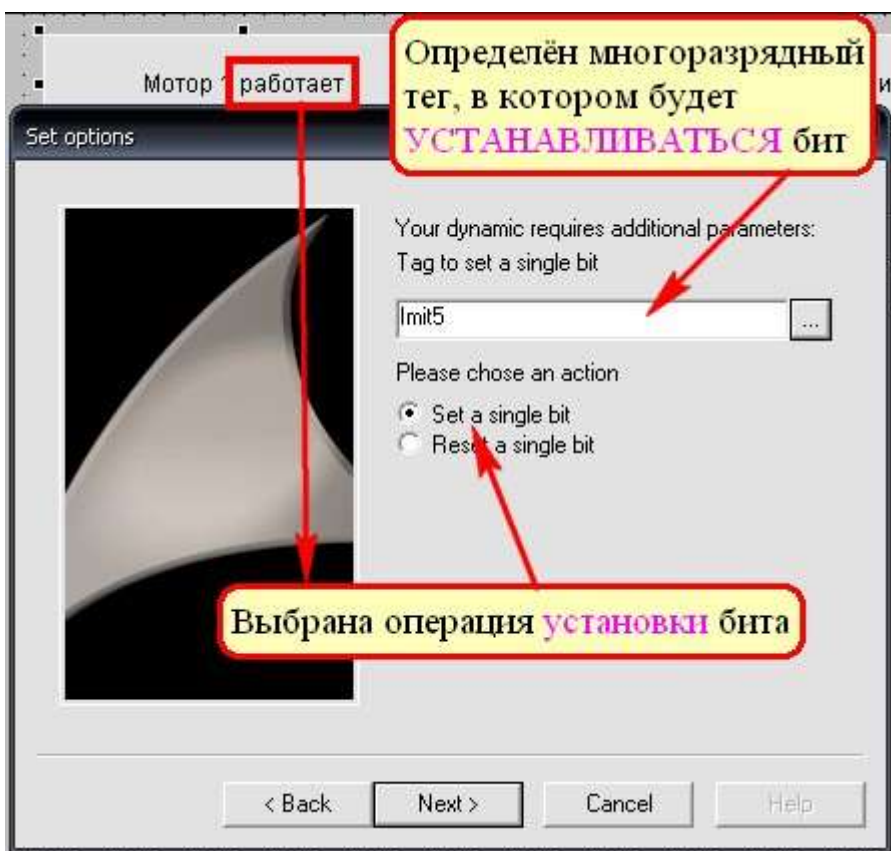


Рис. 3. Выбор типа воздействия на бит тега

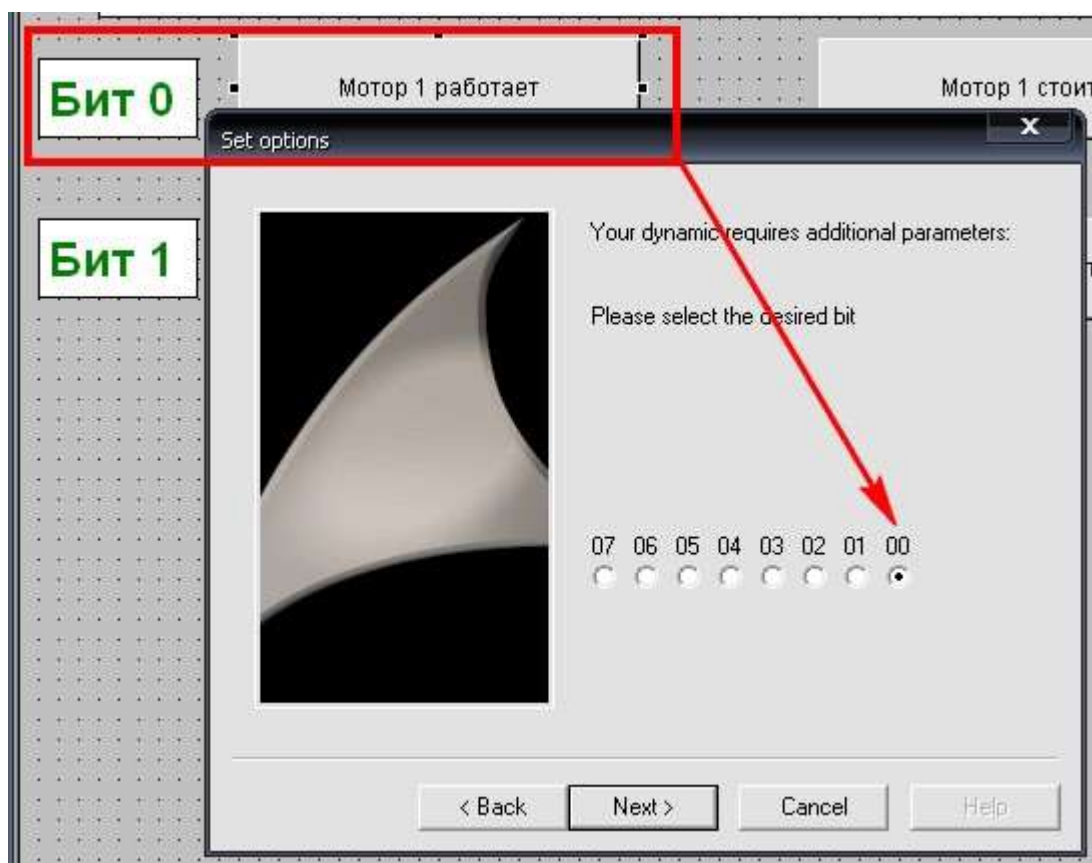


Рис. 4. Выбор бита для связи с кнопкой

В примере для отображения состояния контролируемого тега используется поле ввода/вывода I/O Field, настроенного с помощью Мастера динамики Simultaneous input/output (рис. 5 - 8).

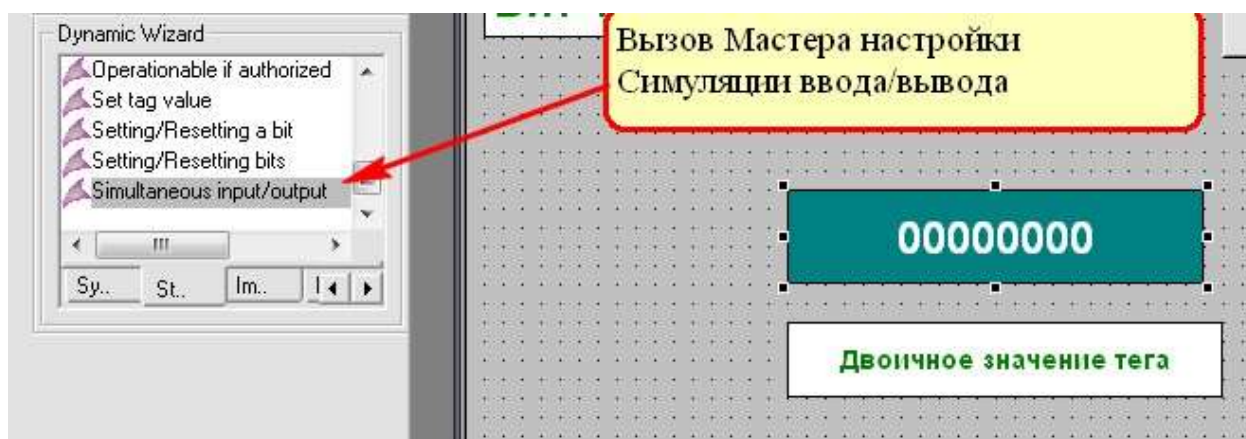


Рис. 5. Переход к мастеру настройки Симуляции ввода/вывода

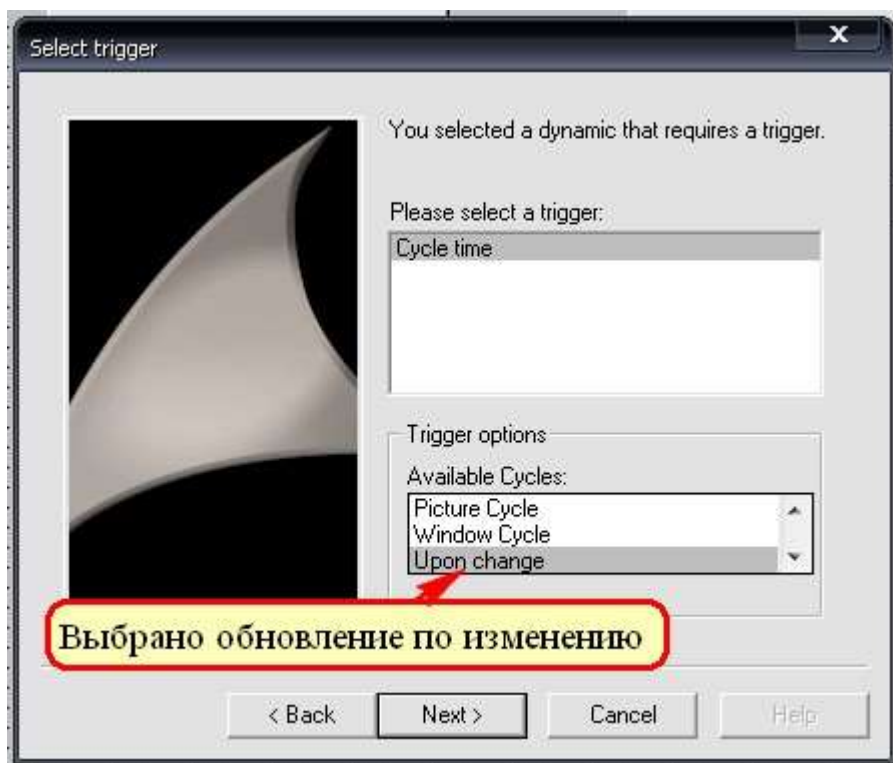


Рис. 6. Выбирается тип обновления картинки

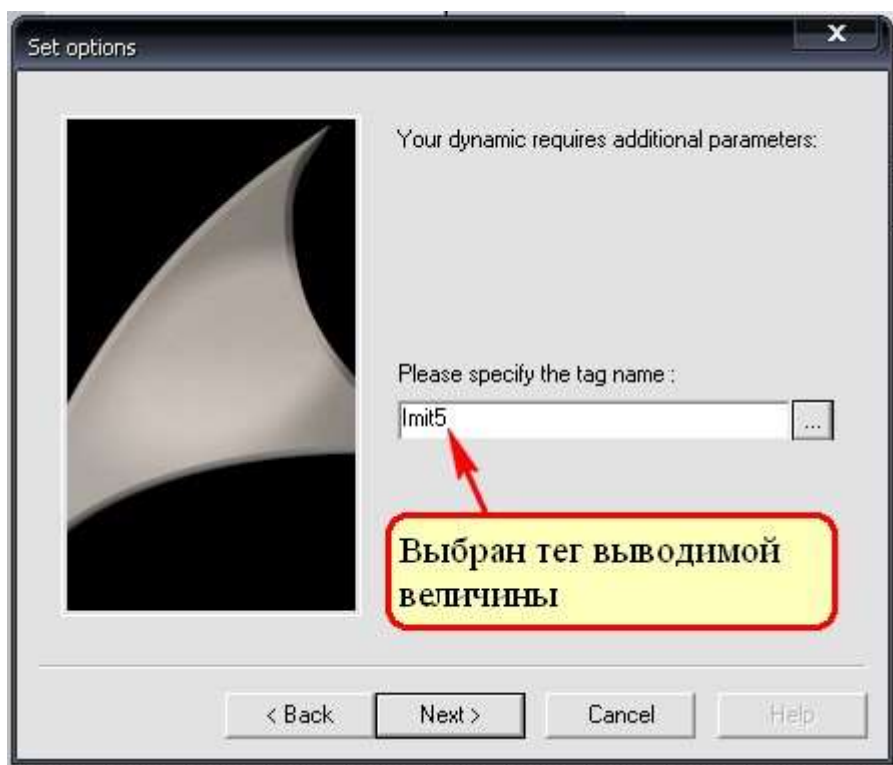


Рис. 7. Привязка контролируемого тега

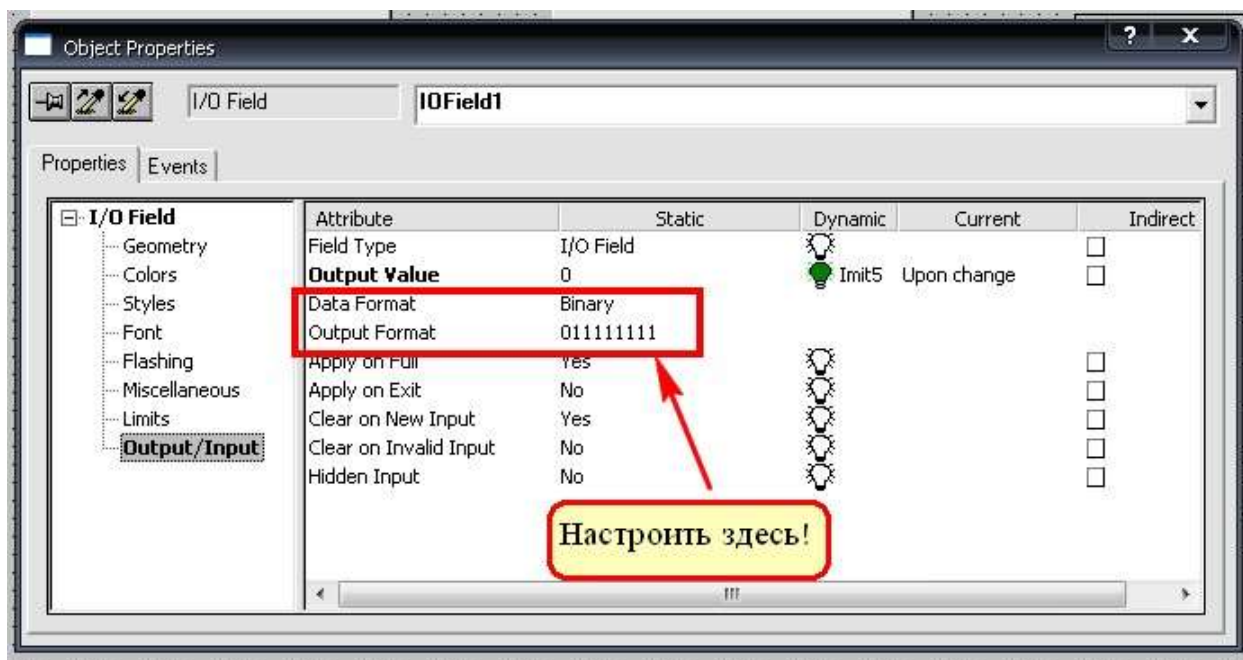


Рис. 8. Настройка вывода состояния тега в двоичном коде

В режиме исполнения при нажатии на кнопки соответствующим образом меняется состояние тега Init5, выводимого в двоичном коде (рис. 9 - 12).

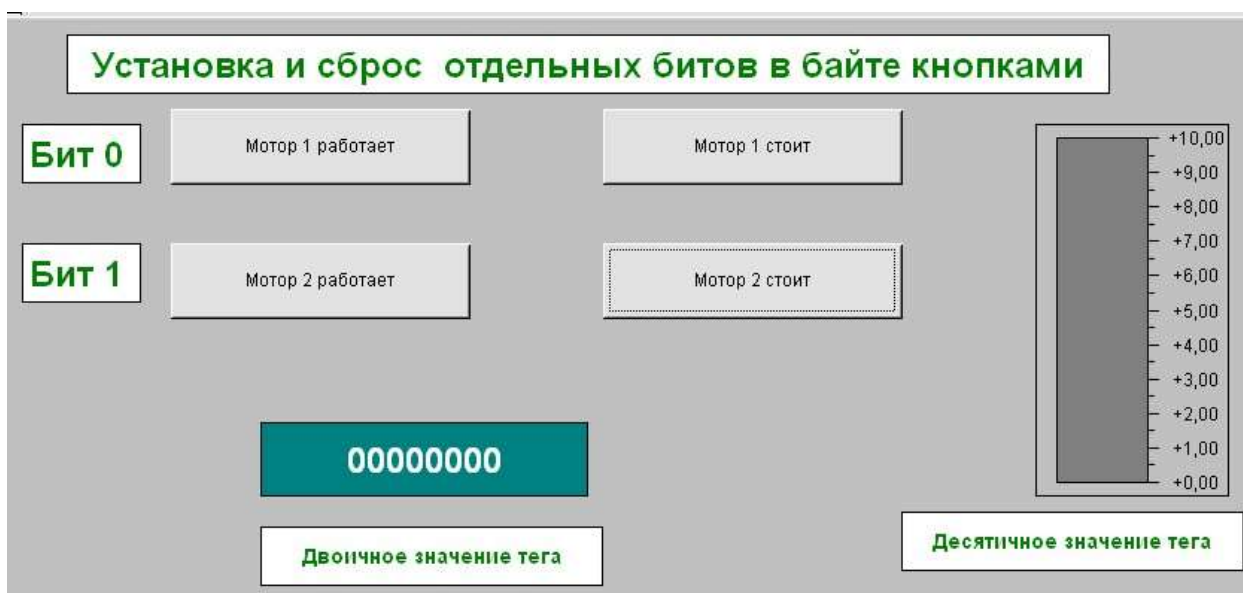


Рис. 9. Режим исполнения – оба двигателя стоят

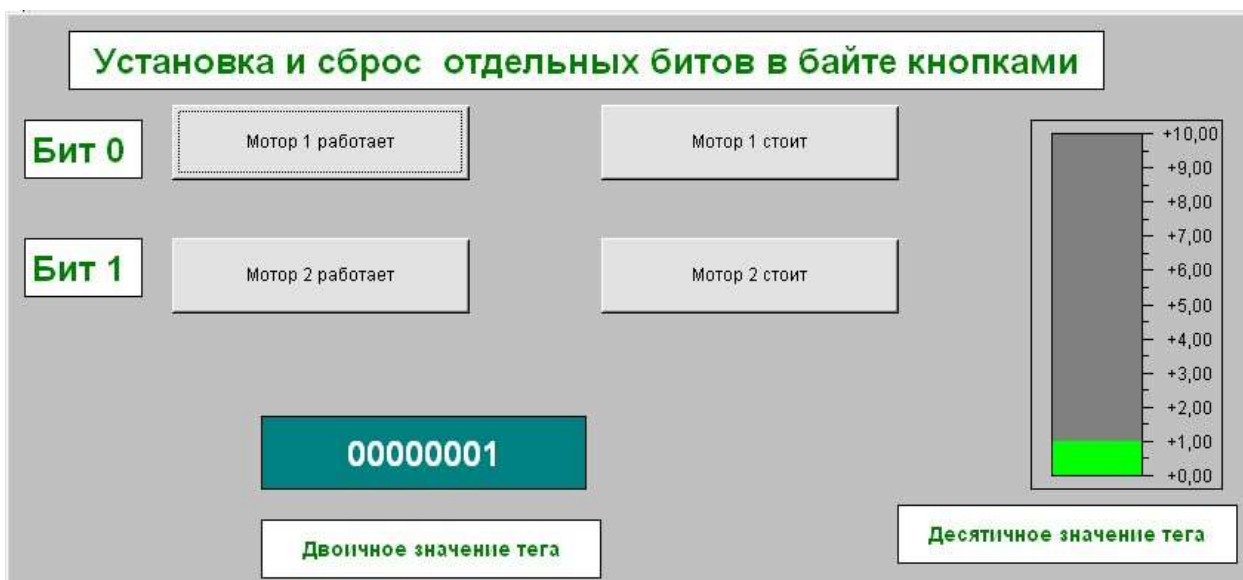


Рис. 10. Режим исполнения – мотор 1 работает

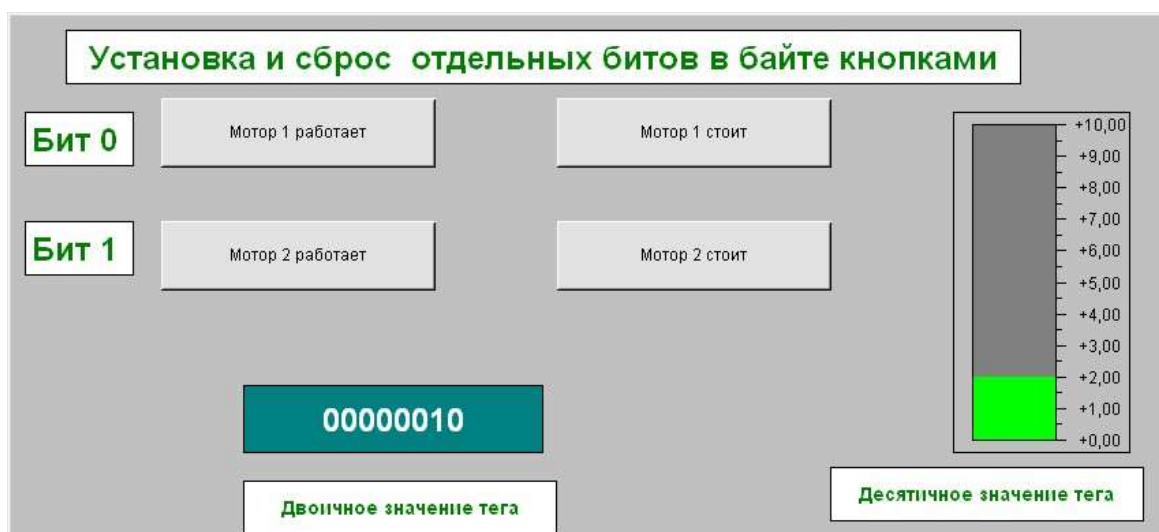


Рис. 11. Режим исполнения – мотор 2 работает

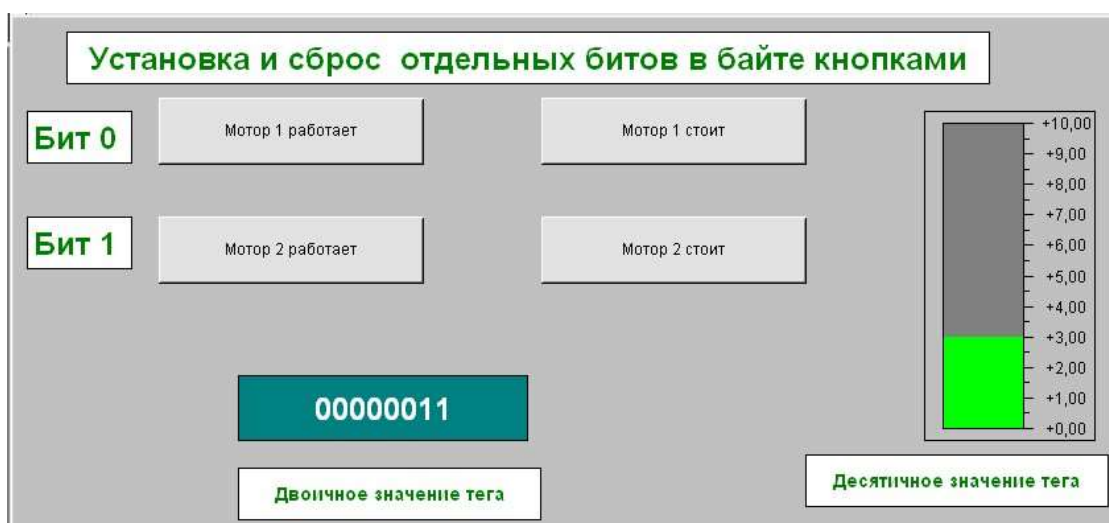


Рис. 12. Режим исполнения – работают оба двигателя