

Задача позиционирования мотора NXT

Введение

На этом уроке Вы научитесь:

1. Пользоваться датчиком положения мотора NXT в программе
2. Управлять двигателем задавая требуемое положение а не мощность
3. Запускать программу использующую NXT, на компьютере
4. Использовать сдвиговые регистры в циклах

Устройство датчика угла поворота в моторах NXT

Фиксация положения мотора

Робототехнике очень часто встречается задача четкой фиксации положения мотора. Для начала мы рассмотрим частную задачу, когда мотор надо зафиксировать в том положении которое было при запуске нашей программы.

Начинаем с создания шаблона программы, с которым мы познакомились на прошлом уроке:



Поскольку после включения программы, NXT не может правильно определить положение энкодера, в начале программы мы должны принять текущее положение энкодера как ноль.

Добавляем на панель диаграмм функцию чтения датчика **NXT I/O -> Read Sensor**, выбираем в выпадающем меню под иконкой функции **Reset Motor** и вставляем ее перед циклом но после функции **Specify NXT**.

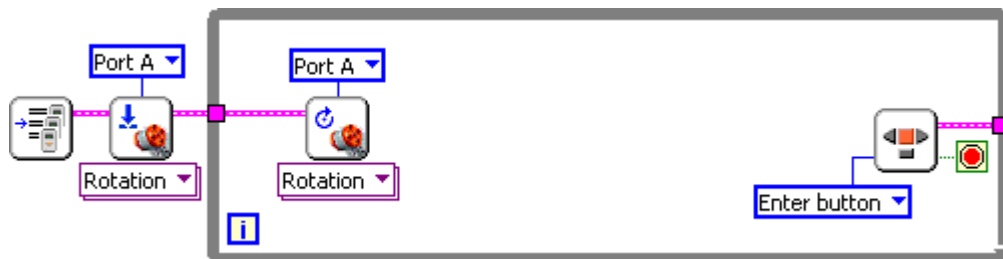
Около верхнего входного терминала функции создаем константу и указываем в ней порт А.



Теперь энкодер готов к работе. Добавляем еще одну функцию чтения датчика **NXT I/O -> Read Sensor**, и вставляем ее внутрь цикла.

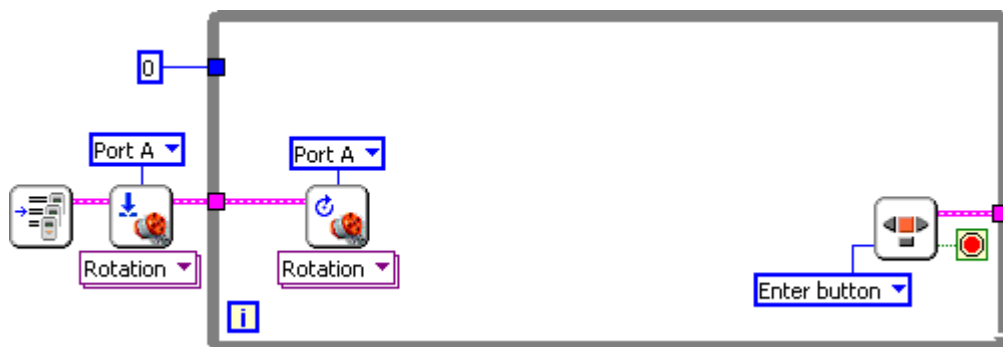


Выбираем в выпадающем меню тип **функции Read Rotation** (прочитать показания энкодера), и указываем константой порт А.



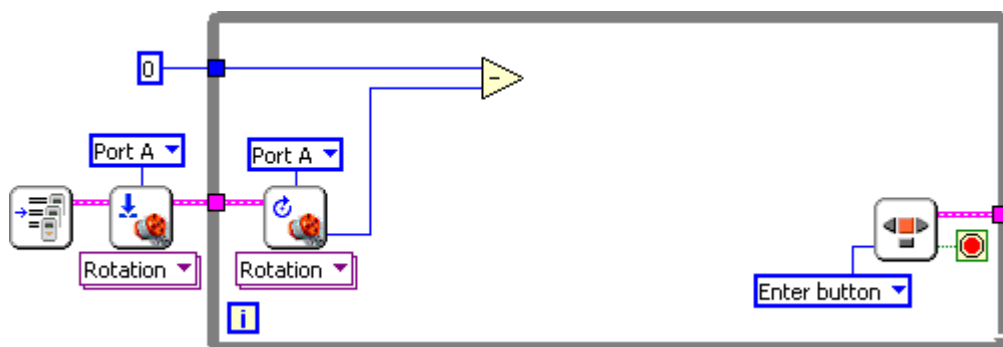
На выходном терминале этой функции мы будем получать актуальное положение мотора.

Создаем числовую константу перед циклом, и заводим ее значение в цикл через туннель.



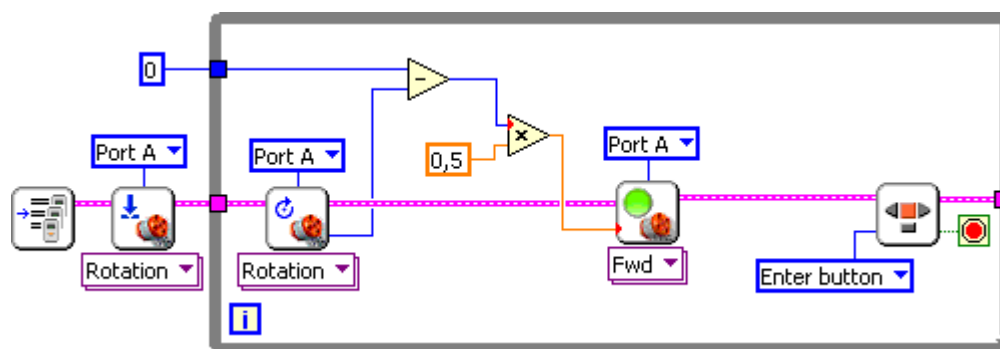
В это константе будет указано положение в градусах, которое мотор должен удерживать, поскольку по условию задачи нам необходимо удерживать начальное положение после запуска программы то мы оставим нулевое значение.

Далее нам нужно сравнить, на сколько реальное положение мотора отличается от заданного положения, для этого вычтем одно из другого. Эту разность дальше мы будем называть ошибкой по положению.

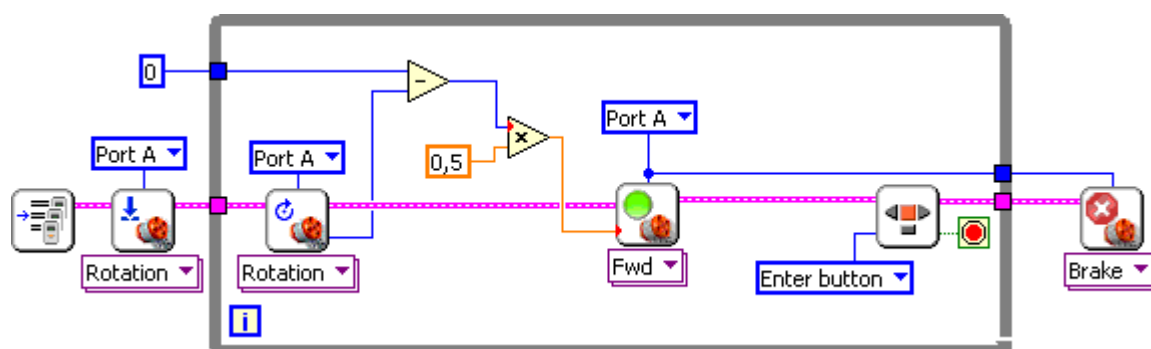


Теперь ошибку по положению умножаем на 0,5.

Добавляем функцию управления мотором **Functions -> NXT I/O -> Motor control**, и выдаем результат умножения как задание мощности на этот мотор.



После цикла вставляем функцию останова двигателя А (**Functions -> NXT I/O -> Motor control** в меню под иконкой выбираем **motor off->brake**).



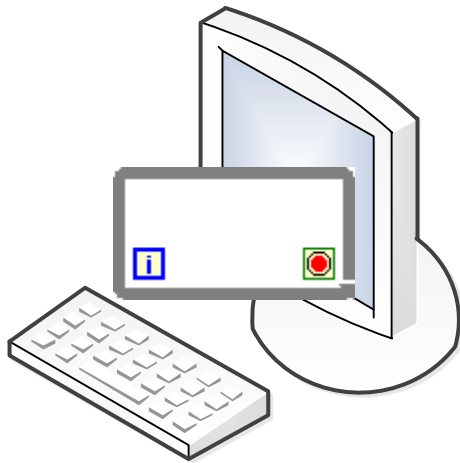
Наша программа готова. Загружаем ее на NXT и проверяем ее работу.

Робот, при запущенной программе ничем крутить не будет, но если вы попытаете повернуть ось двигателя А то у вас не получится это сделать, так как программа автоматически будет противодействовать вашим попыткам повернуть ось мотора.

Поэкспериментируйте с константой 0,5 и сделайте выводы как ее величина влияет на поведение робота при попытке повернуть ось мотора А.

Работа программы на компьютере с использованием NXT

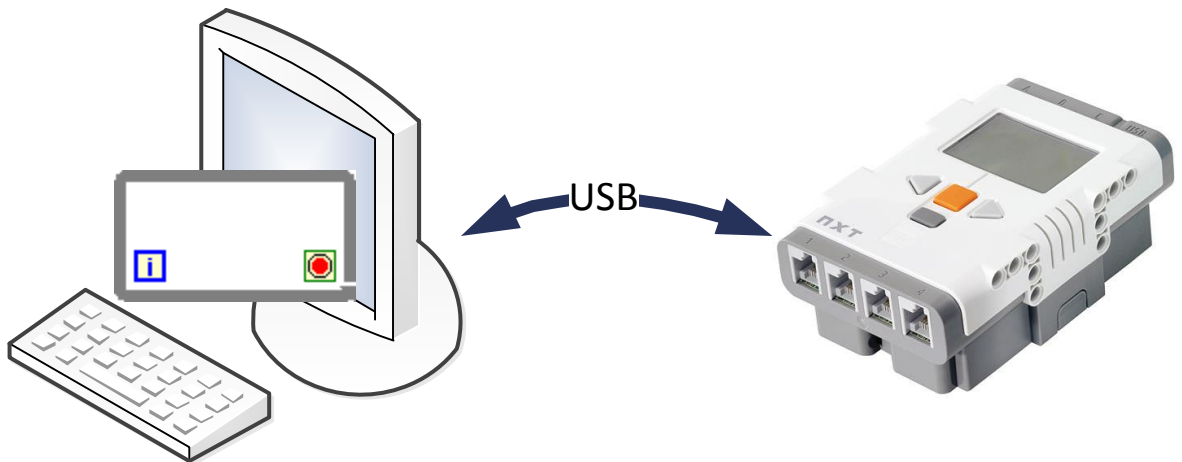
Мы уже научились писать программы которые выполняются только на компьютере или только на NXT.



При выполнении программы на компьютере мы можем использовать лицевую панель для добавления элементов управления, контролов, и индикаторов, для управления программой и для отслеживания ее состояния. А также проводить отладку программы на панели редактирования диаграмм в процессе выполнения программы.

Когда мы записываем программу на NXT то мы лишаемся возможности использовать лицевую панель программы, но зато программа может вести управление моторами и снимать показания с датчиков подключенных к NXT.

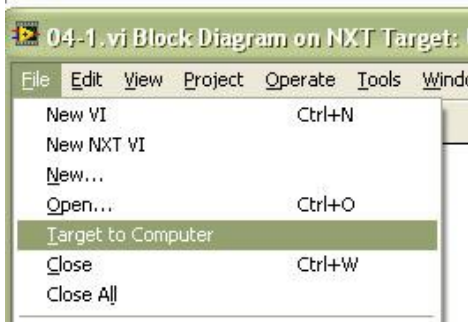
Совместим плюсы обоих способов, и организуем систему следующим образом:



Программа будет выполняться на компьютере, но при этом она будет управлять моторами и получать информацию с датчиков при помощи прямого подключения NXT к компьютеру через USB кабель.

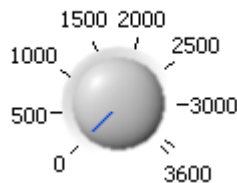
Сохраняем нашу программу, написанную ранее, под другим именем.

Заходим в главное меню и меняем режим привязки программы на Target to computer



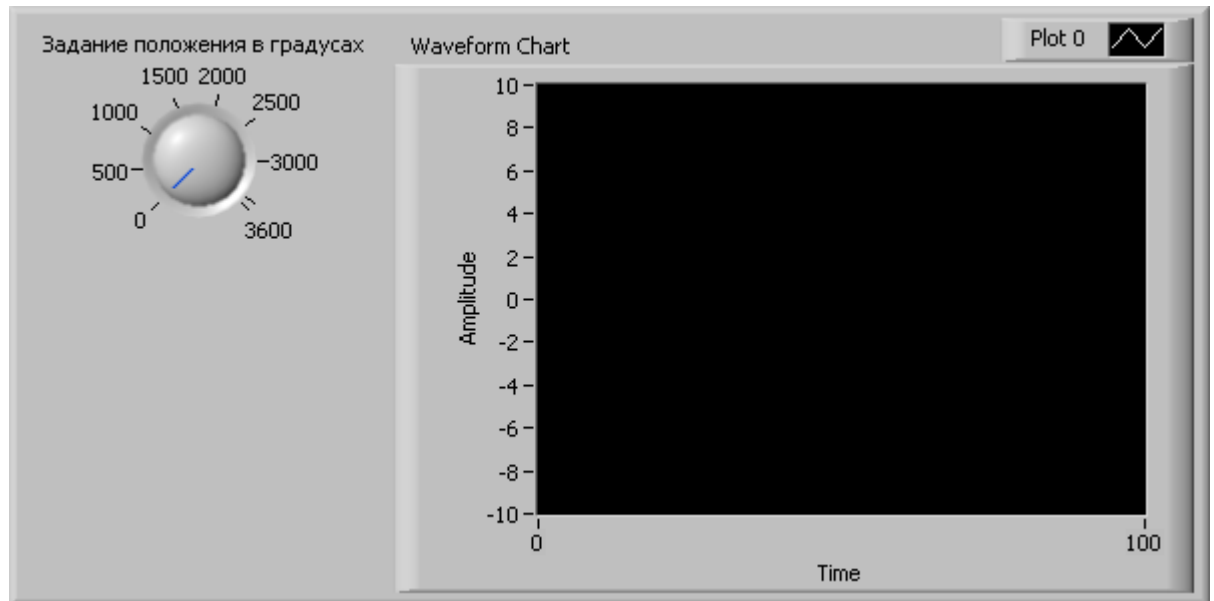
На лицевой панели создаем контрол типа **Dial, NXT Robotics -> Numeric -> Dial**.

Задание положения в градусах



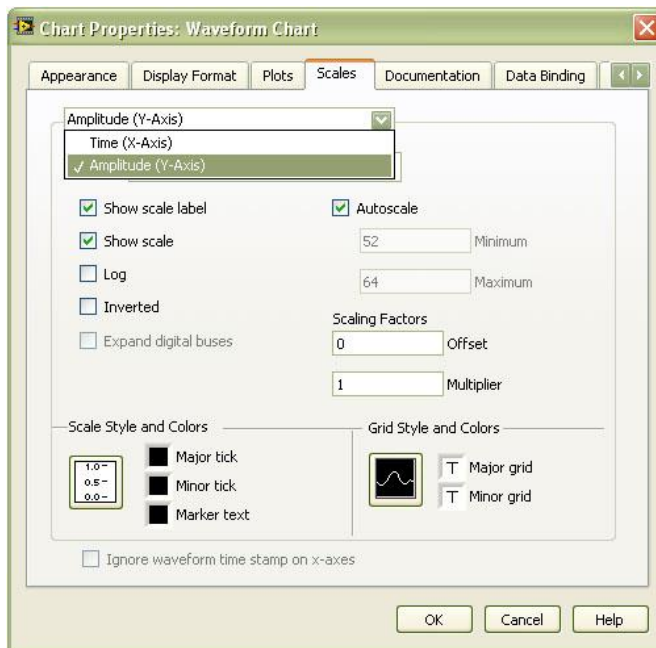
Переименовываем в «**Задание положения в градусах**» и меняем диапазон значений на диапазон соответствующий 10 оборотам двигателя в градусах, то есть от 0 до 3600 (properties -> scale -> scale range [minimum; maximum]).

Еще добавляем на лицевую панель новый для нас, тип индикатора **Тренд (Chart) Graph -> Waveform Chart**.

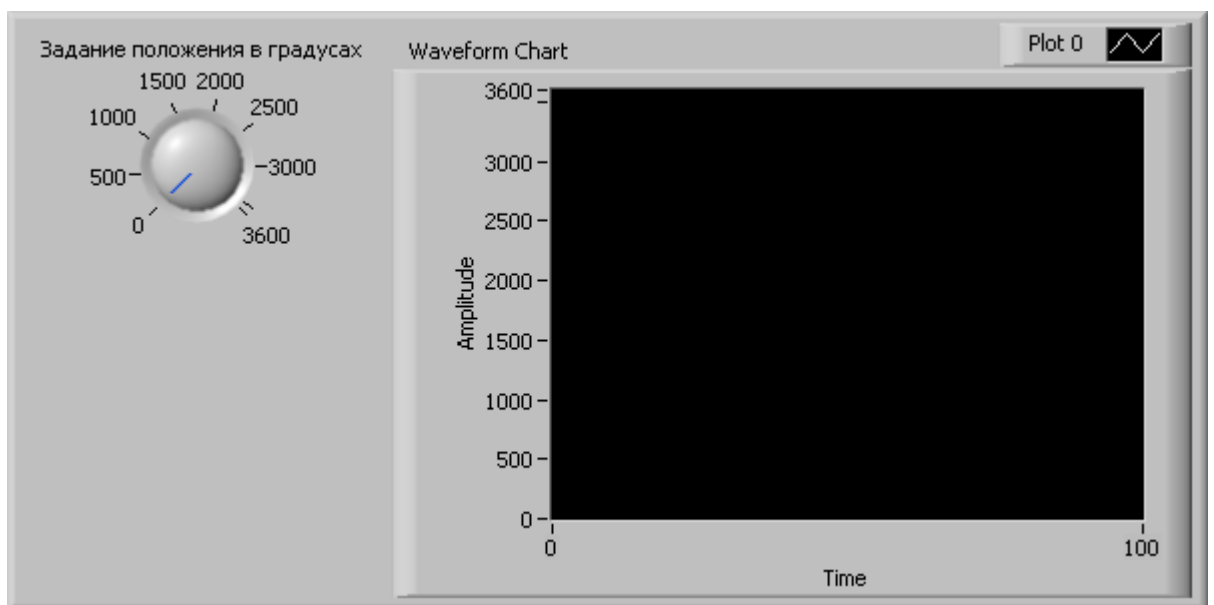


На этом индикаторе будут в виде графиков отображаться задание положения, и реальное текущее положение мотора, потому необходимо настроить шкалу отображения по оси «Amplitude», чтобы мы могли наблюдать изменения значений во всем диапазоне, от 0 до 3600.

Кликаем по **Waveform Chart** правой кнопкой мышки **properties->scales** далее в выпадающем меню выбираем ось-Y (Amplitude (Y-Axe)).

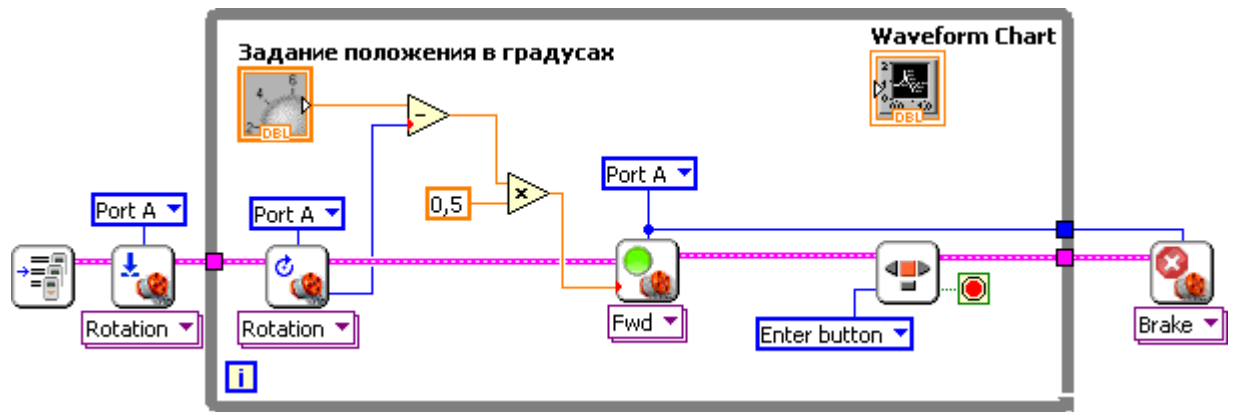



Снимаем галочку Autoscale, а в графы minimum, maximum устанавливаем 0 и 3600.
Нажимаем Ок.

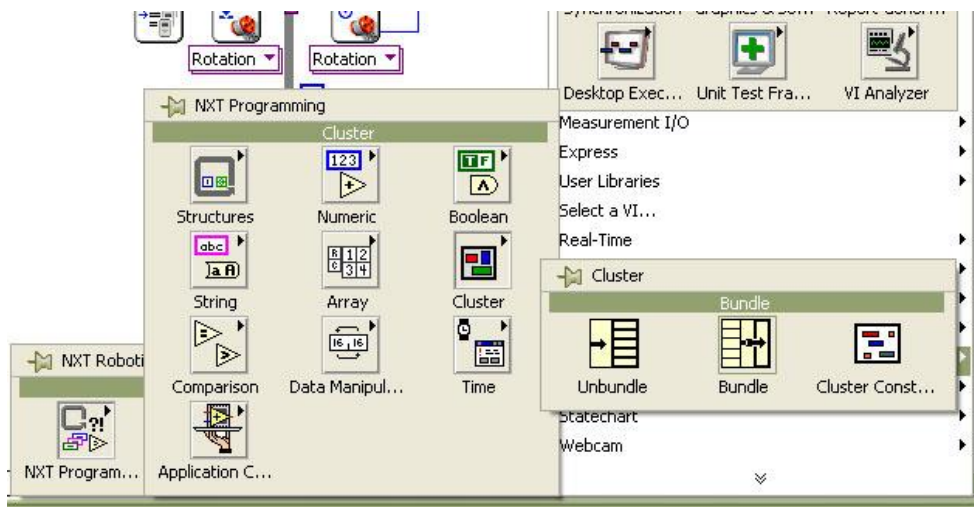


Переходим на панель редактирования диаграмм.

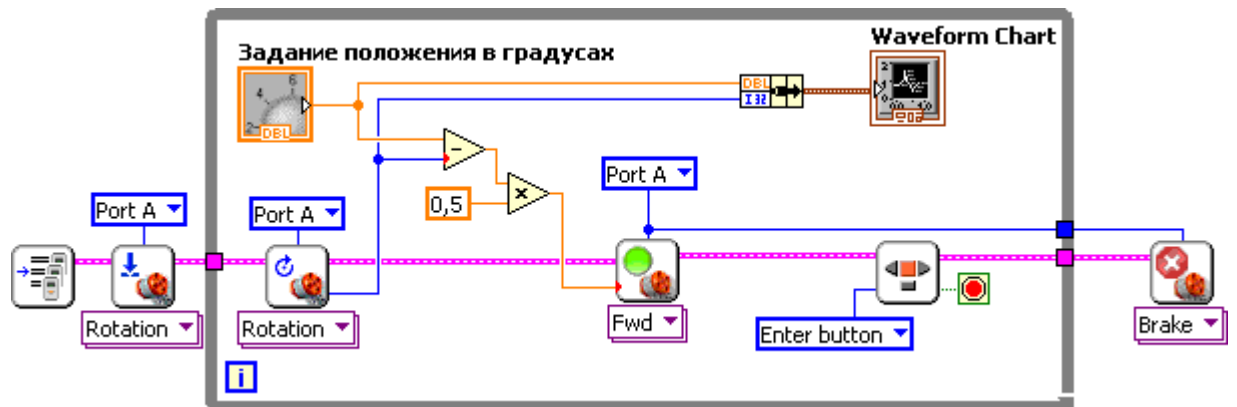
Удаляем константу с заданием положения и туннель ведущий от нее к функции вычитания.
Присоединяем к освободившемуся входу функции вычитания, терминал «задание положения в градусах».



Добавляем функцию **Bundle** (создать массив)  **NXT robotics -> NXT programming -> Cluster -> Bundle**



Присоединяем к первому входу «Задание положения в градусах», ко второму текущее значение энкодера и полученный кластер подаем на вход терминала **Waveform Chart**.



Кластер является структурой, группирующей данные, он может группировать данные разных типов, по аналогии например с реальными проводами и объединением их в кабель.

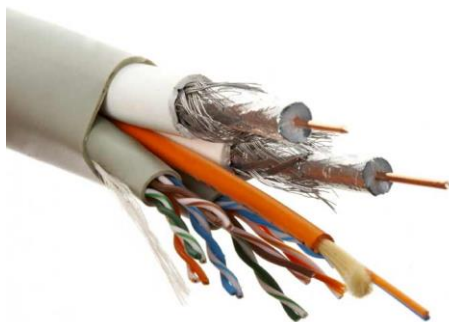


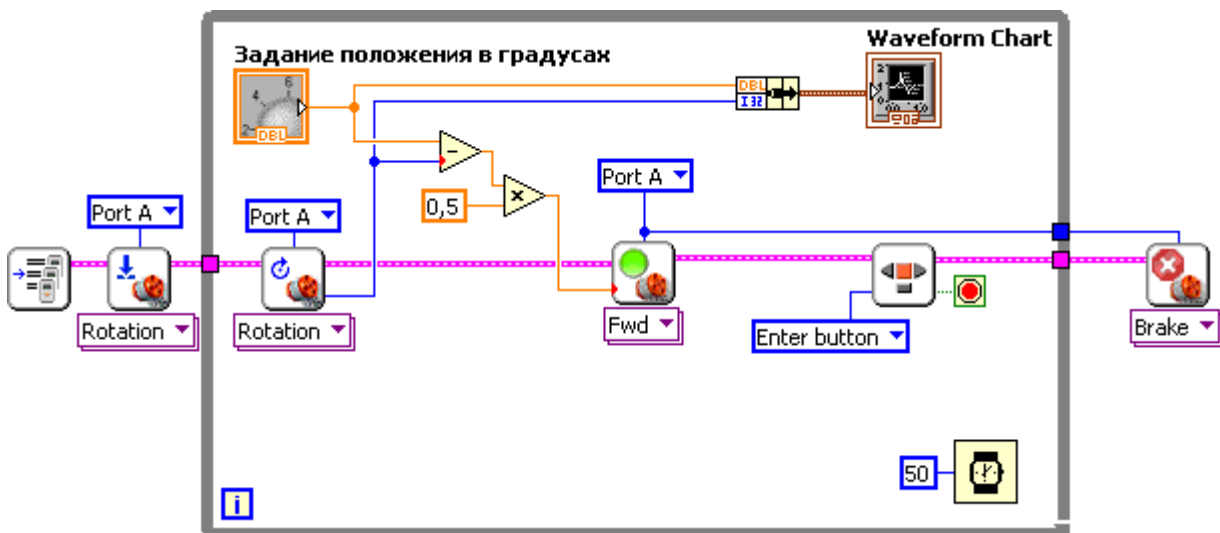
Рисунок 1 Кабель с проводами - кластер с данными



Рисунок 2 отдельные провода - отдельные связи с данными

Более подробно мы будем знакомиться с кластерами в последующих занятиях.

Теперь добавляем временную задержку в цикл, для разгрузки центрального процессора, величину задержки выставим в 50 мс.



Запускаем программу.

Изменяя задание контролем на лицевой панели мы можем наблюдать на графике, как на сколько быстро мотор переходит в заданное положение.

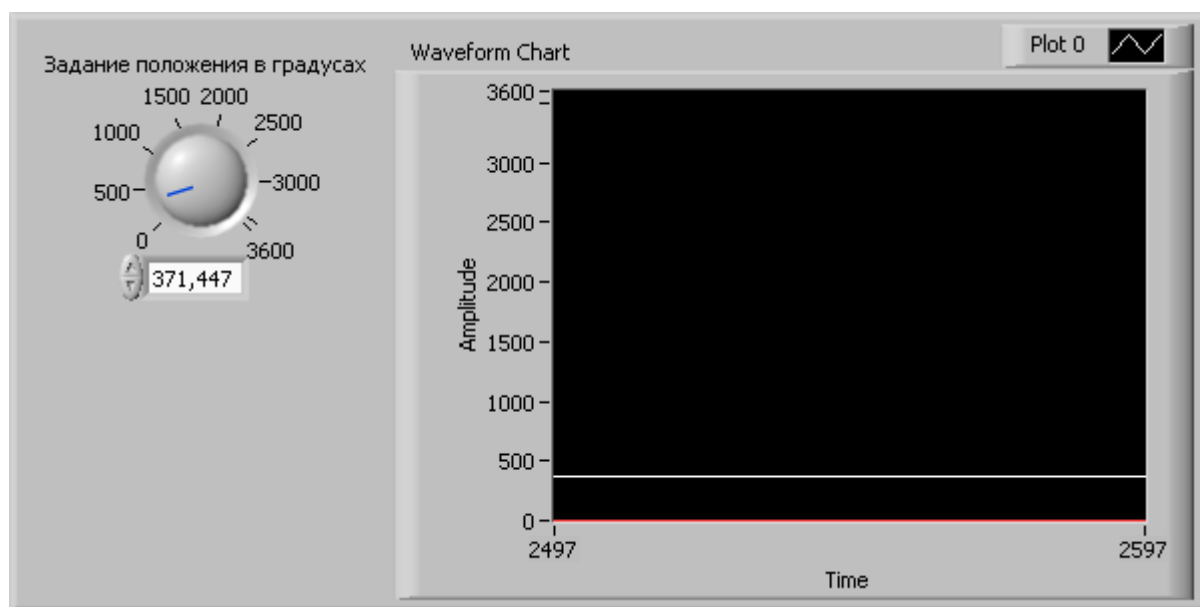
Эксперимент

Проведем эксперимент, как влияет константа 0,5 на скорость перехода в требуемое положение и на точность этого перехода.

На контроле «Задание положение в градусах» кликаем правой кнопкой мышки и выбираем **Visible Items -> Digital display**. Рядом с ручкой контрола появится дисплей, на котором будет дублироваться значение, но в численном виде, также через него можно задавать значение как мы это делали через обычный **Numeric control**.



Для удобства перемещаем дисплей под рукоятку управления



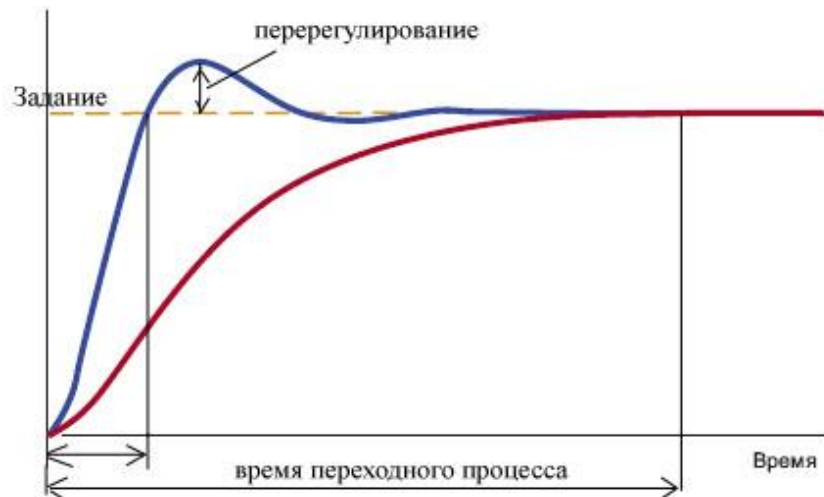
Теперь мы можем мгновенно менять задание изменяя значение в окошке дисплея.

Эксперимент будет проходить циклически, для каждого значения константы из таблицы, далее будет приведена таблица со значениями и описан алгоритм проведения эксперимента:

Пропорциональный Коэффициент усиления (Значение константы)	0,2	0,5	0,8	1	1,5	2	5	8	8	10
Время переходного процесса										
Перерегулирование										

1. Устанавливаем новое значение константы
2. При помощи дисплея на лицевой панели устанавливаем задание равное 0
3. Запускаем программу
4. Набираем в дисплее 1800 градусов – это соответствует 5 оборотам двигателя
5. Останавливаем программу когда двигатель остановится
6. Измеряем время когда было выдано задание и когда мотор переместился в соответствии с этим заданием

7. Находим разность этих значений, и записываем в ячейку таблицы «Время переходного процесса»
8. Измеряем по графику перерегулирование, это то на сколько мотор проскочил задание



Синий график – перерегулирование есть измеряем и записываем в таблицу, красный график – перерегулирования нет, в таблицу записываем 0.




9. Возвращаемся к 1 пункту и повторяем все действия.

Заключение

В заключении этого урока, мы немного «причешем» программу.

1. Терминалы контролов и индикаторов на панели редактирования диаграмм изначально после создания имеют большой размер, для экономии места их можно уменьшить.

Кликаем правой кнопкой на терминал и снимаем галочку **View As Icon** после этой

операции терминал уменьшится в размерах  ->  -> 

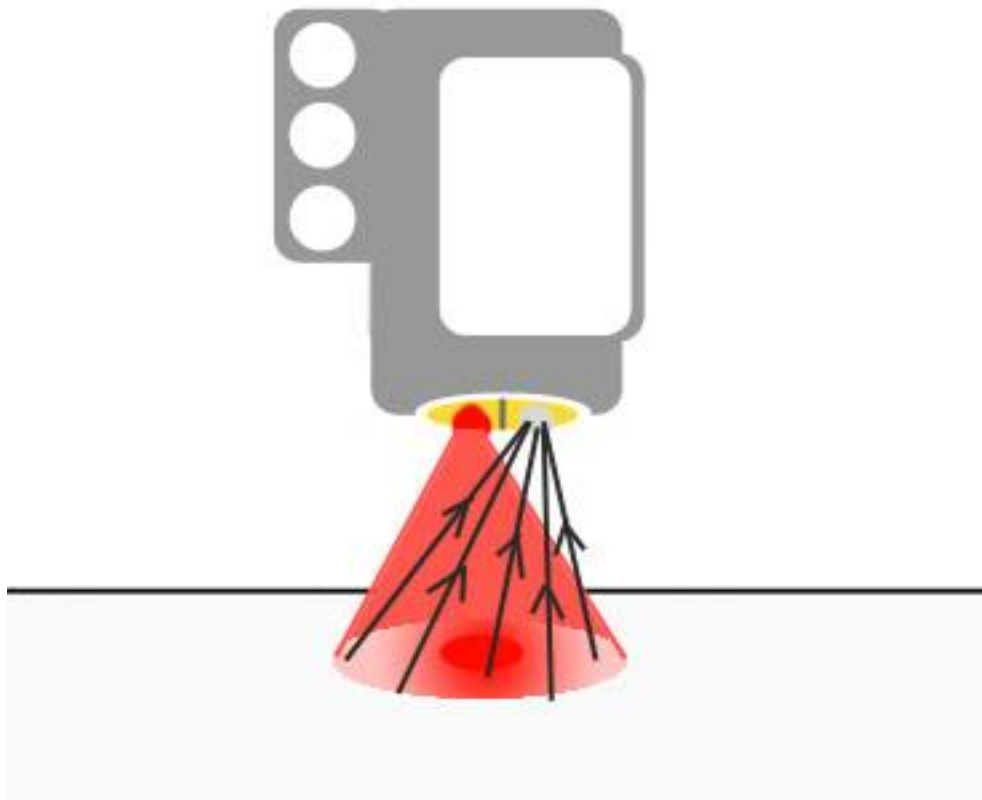
2. Любая константа в программе должна иметь комментарий, поясняющий ее назначение. В нашей программе только одна константа, кликаем на ней правой кнопкой мышки и в меню выбираем **Visible Items -> Label**. Рядом с константой появится окошечко с ее названием, в котором мы пишем латинскими буквами «Kp»

Kp


. «K» – означает что «коэффициент усиления», а «р» - «пропорциональный», это следует запомнить, так в дальнейшем мы будем очень часто называть константы.

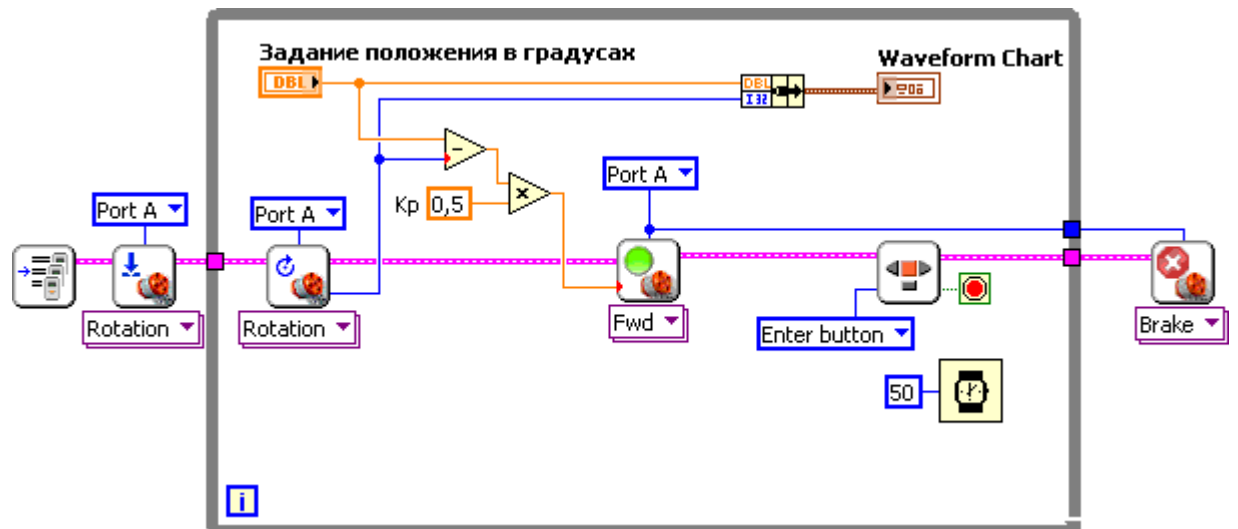
Перетаскиваем название константы к левой стенке Kp .

Теперь наша программ



следующий вид

а приняла



На этом мы заканчиваем урок.