Ярмолинский Леонид Маркович

УРОК 2. ПЕРВАЯ ПРОГРАММА ДЛЯ NXT НА LABVIEW

ВВЕДЕНИЕ

В прошлой статье мы познакомились с интерфейсом Labview и написали свою первую программу. На этом занятии мы создадим несколько программ для управления двухмоторным роботом с помощью контроллера Lego NXT. Для этого нам понадобится сам робот и кабель для записи программы на контроллер. Давайте начнем.

ПРОГРАММА «ПРЕРЫВИСТОЕ ДВИЖЕНИЕ ВПЕРЕД»

Первая программа будет служить для передвижения робота вперед с периодическими остановками.

Создаем новую программу, открываем панель редактирования диаграмм и в главном меню выбираем режим привязки к **NXT file → Target to NXT** (рис. 1).

Обе панели немного изменят свой внешний вид, в левом нижнем углу появится оранжевое поле статуса привязки к NXT, а на линейке инструментов появится пара новых кнопок (рис. 2):

– запись программы в память контроллера NXT;

– запуск программы на контроллере в режиме отладки аналогично кнопке .

Также стоит отметить, что так как программа теперь выполняется не на компьютере, а на контроллере, то при нажатии кнопки , программа будет переписана на контроллер, после чего автоматически запустится. Об этом нужно помнить, если вы работаете с мобильным роботом, чтобы избежать аварии, например падения робота со стола.

Палитры контролов и функций также изменились, а точнее, остались только те функции, которые NXT способен выполнить.

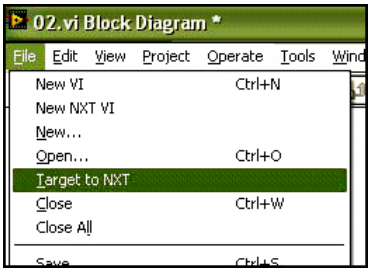


Рис. 1

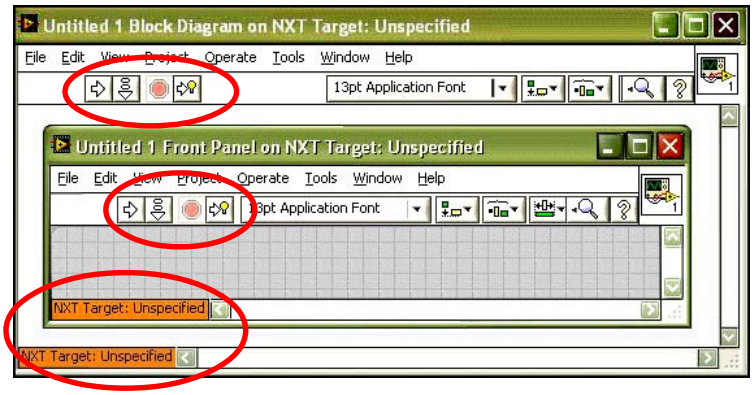


Рис. 2

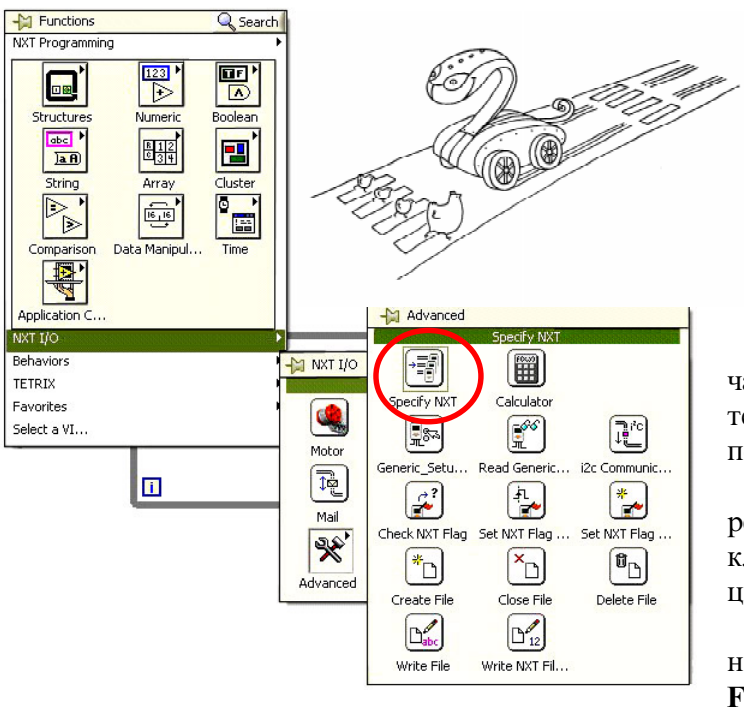


Рис. 4

Рис. 3

Перейдем на панель редактирования диаграмм и начнем писать программу.

Создадим основной цикл программы While Loop (рис. 3).

Терминал прерывания оставляем неподключенным.

Любую программу, выполняемую на NXT, мы будем начинать с функции **Specify NXT**  **Functions → NXT I/O → Advanced → Specify NXT**, основная ее задача сформировать кластер с параметрами, который мы будем подключать ко всем функциям при работе с различными устройствами, подключенными к NXT (рис. 4).

Добавляем **Specify NXT** перед циклом и заводим выходной кластер этой функции NXT в цикл через туннель (рис. 5)

Добавляем функцию управления мощностью мотора **Functions → NXT I/O → Motor control** (рис. 6).

Кликаем правой кнопкой мыши на верхний входной терминал функции и в появившемся меню выбираем create constant. По умолчанию в созданной константе будет значение «All ports», оно нас устраивает, поэтому мы его не меняем.

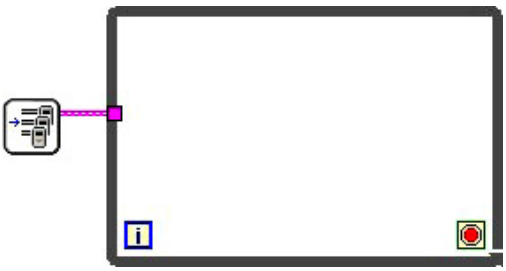


Рис. 5

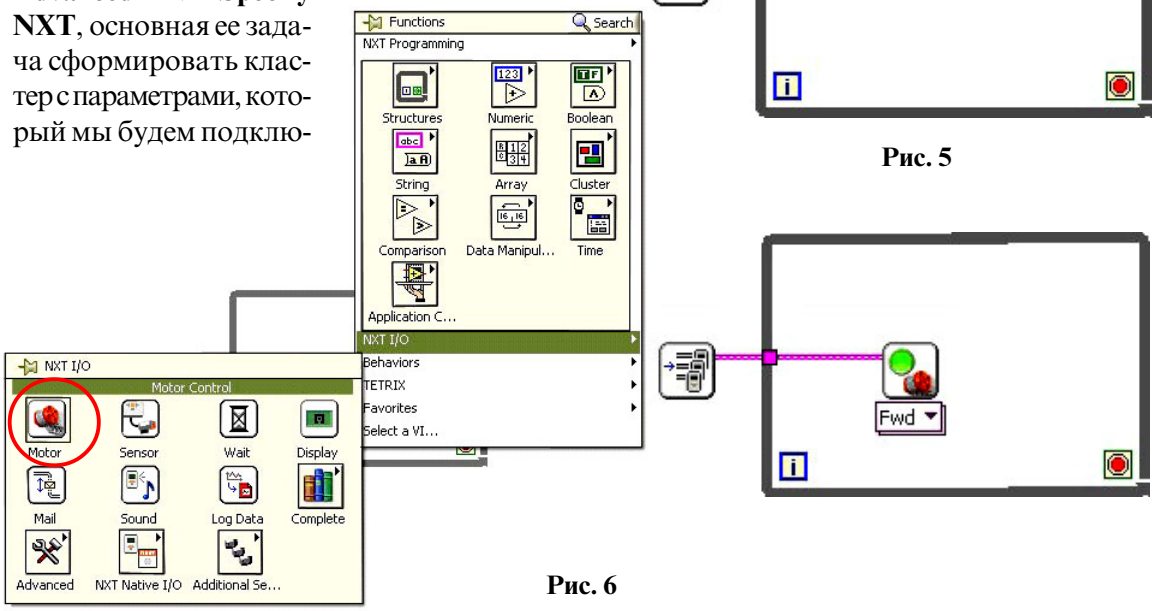


Рис. 6

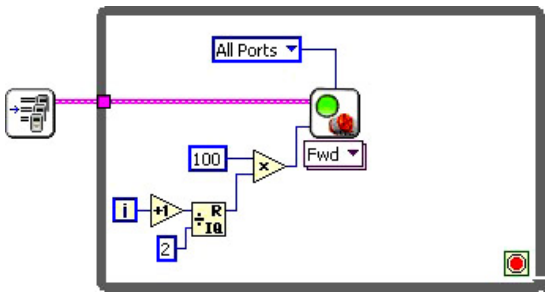


Рис. 7

К терминалу задания мощности подключим выражение, зависящее от номера текущей итерации цикла(рис. 7).

К текущей итерации *i* прибавляем 1, потом берем остаток от деления на 2 и умножаем на 100, таким образом, получается, что на любой четной итерации остаток от деления на 2 будет 0, и, следовательно, подаваемая на вход моторов мощность будет равна 0 (0·100 = 0), а когда итерация имеет нечетный номер, остаток от деления на 2 будет

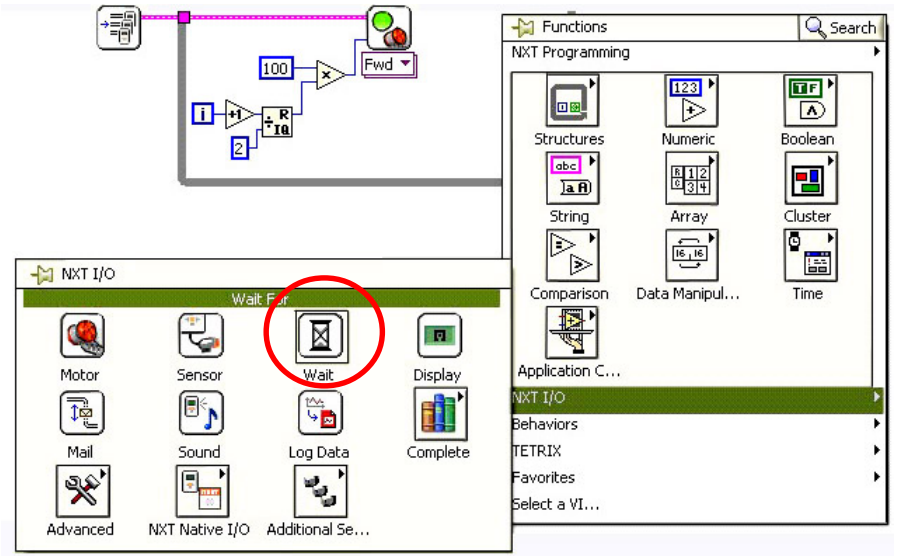


Рис. 8

равен 1, и мощность будет равна 100 (1·100=100).

После задания мощности на моторы вставляем временную задержку **Functions → NXT I/O→ Wait for** (рис. 8).

И при помощи числовой константы устанавливаем значение задержки, равное 1 секунде (рис. 9).

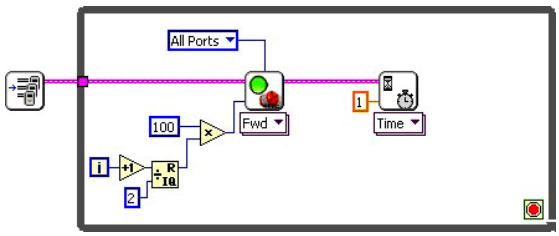


Рис. 9

Теперь дело дошло до терминала прерывания цикла, который, в начале занятия мы оставили без внимания. Добавляем функцию чтения состояния кнопок на **NXT Functions → NXT I/O → Complete → Sensors → Read NXT Buttons** (рис. 10).

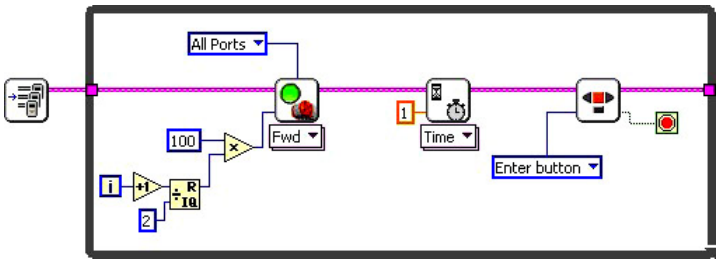


Рис. 10

Кликая правой кнопкой мышки на входном терминале

выбора кнопки, создаем константу (Create constant) и оставляем значение «Enter Button», что соответствует кнопке в виде оранжевого квадрата. Выходной терминал со считаным состоянием кнопки подключаем к терминалу прерывания цикла.

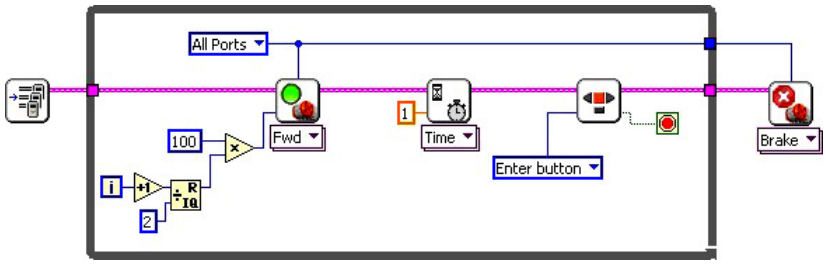


Рис. 11

Наша программа почти готова, осталось только выключить моторы после завершения цикла. Для этого добавляем уже известную нам функцию **Functions → NXT I/O → Motor control** после цикла и в меню под иконкой выбираем **motor of → brake**, также не забываем подключить константу «All Ports», чтобы команда выдалась на все порты (рис. 11).

Сохраняем программу на компьютере.

Загружаем программу на NXT при помощи кнопки **Deploy** на линейке инструментов, дожидаемся загрузки, отключаем робота, и проверяем, как работает программа.

Поздравляю! Наша первая программа для NXT работает!

ПРОГРАММА «РЕЛЕЙНАЯ ЗМЕЙКА»

Теперь переделаем задание мощности, чтобы робот ехал не прерывисто с интервалом в одну секунду, а змейкой с таким же интервалом. Для этой задачи нам потребуется раздельное управление мощностью левого и правого мотора.

Добавляем еще одну иконку управления мотором **Functions → NXT I/O → Motor control** и указываем, что она выдает команду мотору, подключенному к порту C («Port C», а у той, которая выдавала команды на все порты, указываем только порт A («Port A». Также создаем отдельную константу со значением «All ports» для функции «Brake», которая следует после цикла (рис. 12).

В предыдущей программе при помощи остатка от деления номера текущей итерации на 2 мы получали чередующиеся 0 и 1, удалим последующее умножение на 100 и подадим этот переменный сигнал на вход структуры **case** (рис. 13).

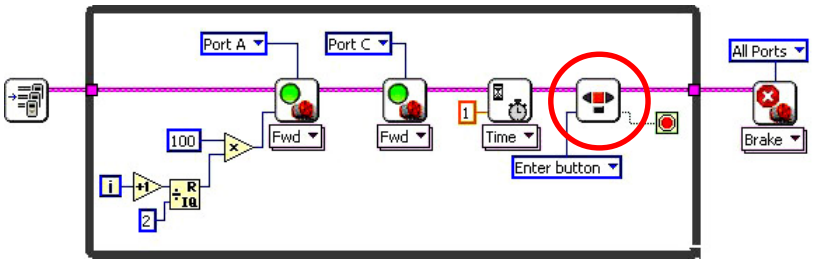
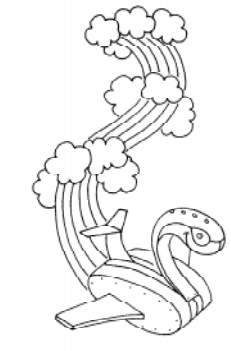


Рис. 12



Выбираем окно структуры **case** для значения 0. Создаем две

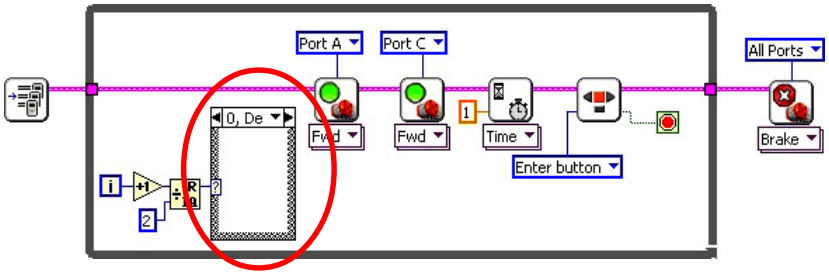


Рис. 13

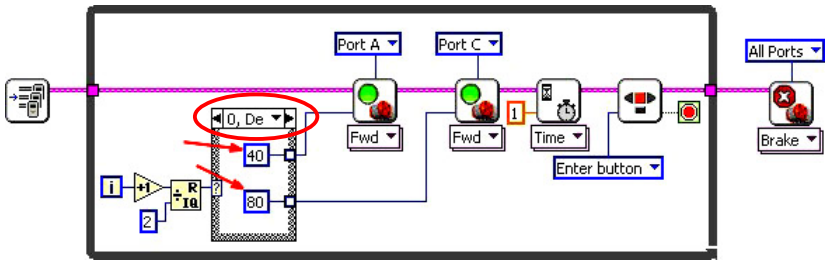


Рис. 14

числовые константы со значениями 40 и 80 и подключаем их к входам задания мощности моторам (рис. 14).

Выбираем окно структуры **case** для значения 1 и проделываем ту же операцию, только значения 40 и 80 меняем местами, а подключение к входам задания мощности производим через появившиеся выходные туннели структуры **case** (рис. 15).

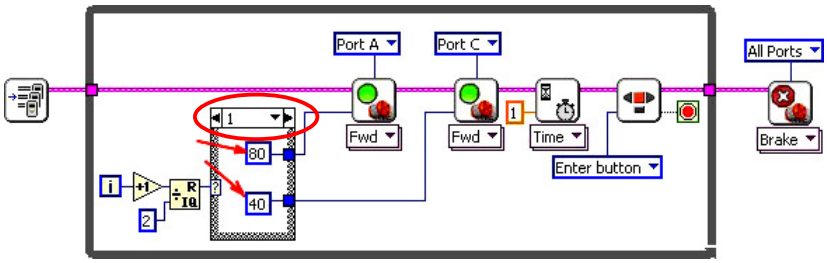


Рис. 15

Сохраняем программу на компьютере.

Загружаем программу на NXT при помощи кнопки **Deploy** на линейке инструментов, дожидаемся загрузки, отключаем роботи проверяем, как работает программа.

ПРОГРАММА «ПЛАВНАЯ ЗМЕЙКА»

В предыдущей программе из одного состояния в другое робот переходил резко, рывком, то есть как только начиналась новая итерация цикла, задание мощности на мотор менялось мгновенно на 40 процентов от полной мощности мотора. Сейчас мы напишем программу, которая плавно меняет задание мощности за счет использования новых функций и за счет уменьшения длительности каждой итерации цикла.

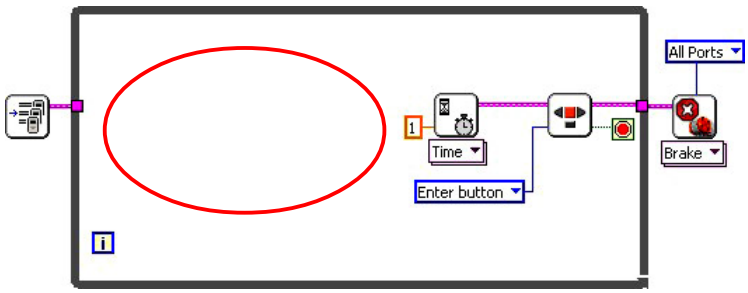


Рис. 16

Сохраняем открытую программу под новым именем и удаляем часть содержимого, чтобы получилась следующая заготовка (рис. 16).

Кликаем на меню выбора под иконкой **Wait for time** и выбираем в нем режим зада-

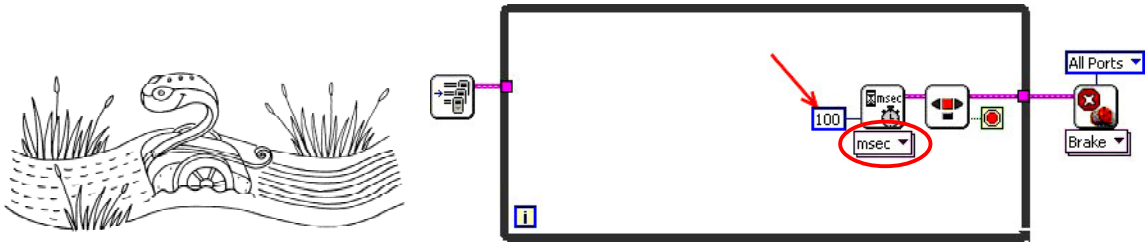


Рис. 17

ния времени в миллисекундах **Wait for time → msec**.

Указываем новую длительность задержки, равную 100 миллисекундам (рис. 17).

Теперь за одну секунду у нас будет выполняться 10 итераций цикла.

Добавляем функцию рулевого управления 2 моторами **steering on**  , **NXT Functions → NXT I/O → Complete → Motors → Steering On**.

Эта функция автоматически выдает задание мощности на 2 выбранных мотора.

Она имеет три синих входных терминала. К верхнему подключается указание, какой парой моторов мы хотим управлять. Создаем константу на входе этого терминала и указываем моторы A и C («Ports A&C»). Оставшиеся два отвечают за выдаваемую мощность на двигатели и радиус поворота. Поскольку предполагается, что робот будет двигаться равномерно, то мощность мы зададим в виде константы со значением 50 (рис. 18).

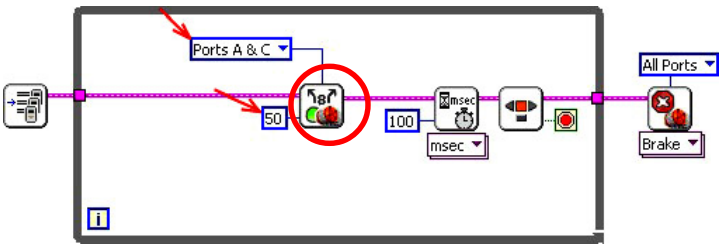


Рис. 18

Мы подобрались к самому интересному: как было сказано выше, вход функции **Steering on** отвечает за радиус поворота. Диапазон подаваемых значений на этот вход лежит в пределах от –100 до 100. Знак числа определяет в какую сторону будет поворачивать, а величина – насколько резкий будет поворот, то есть при значении 0 робот будет ехать прямо, а при 100 или –100 будет крутиться на месте в одну или в другую сторону.

Подадим на этот вход функцию синуса, для плавного изменения радиуса поворота робота.

Добавляем функцию синуса  **Numeric → Trigonometry → Sine**.

На вход функции синуса подадим номер итерации поделенный на 10. А выход, умножив на 35, подадим на входной терминал steering функции **Steering on** (рис. 19).

Сохраняем программу на компьютере.

Загружаем программу на **NXT** при помощи кнопки **Deploy** на линейке инструментов, дожидаемся загрузки, отключаем робот и проверяем, как работает программа.

На этом наше занятие подошло к концу, в заключение поэкспериментируйте со значениями всех констант, используемых в этой программе, сделайте выводы, как эти параметры влияют на поведение робота.

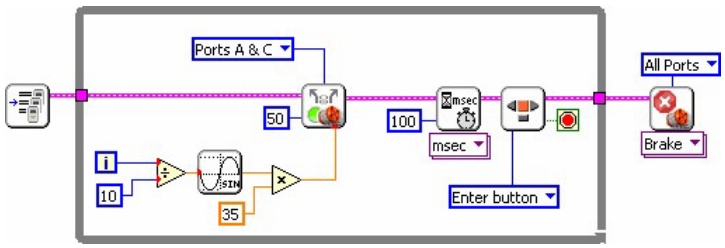


Рис. 19

Ярмолинский Леонид Маркович,
ведущий инженер-программист
ООО «ПромАвтоматика»,
педагог дополнительного образования
ГБОУ СОШ № 255
(кружок робототехники).