

Перевод отрывка статьи Джеффа О'ври «Программирование робота Alpha Rex для обхода препятствий»

Автор: Jeff O'wrey

Перевод: С.С. Химка

Источник: Lego Mindstorms NXT paper

Для создания нашей программы движения воспользуемся одним из важнейших свойств объектно-ориентированного программирования – абстракцией. В общем виде суть ее состоит в том, что она позволяет смотреть на объект, не заставляя себя разобраться в той совокупности сложных частей из которых состоит данный объект. Применительно для нашей задачи абстракция заключается в следующем: сперва создаются элементарные блоки, такие как блок наклонов робота из стороны в сторону и блок движения ноги вперед назад. Второй уровень предполагает создание усложненных блоков, позволяющих роботу ходить вперед, назад, поворачивать вправо и влево. На третьем уровне из описанных блоков строится программа, при выполнении которой робот ходит по квадрату или огибает препятствия.

ПОКАЧИВАНИЯ РОБОТА ИЗ СТОРОНЫ В СТОРОНУ

Для выполнения движения робота из стороны в сторону используется серводвигатель, находящийся на левой ноге (подключенный в разъем С). Использование серводвигателя подразумевает, что основная функция этого мотора переместить, или привести в действие рычаг на ограниченное расстояние. Перемещение на ограниченное расстояние означает, что двигатель не вращается неограниченное время, а делает ровно столько вращений, чтобы переместить рычаг или соединить механические компоненты. В нашем случае эти компоненты – ноги робота, которые имеют две степени свободы: перенос веса с одной ноги на другую,

и движение одной ноги вперед одновременно с движением второй ноги назад.

Наклоны робота из стороны в сторону могут быть разделены на девять отдельных случаев действия серводвигателя С, необходимых, чтобы осуществить основное движение. В таблице 3.1 перечислены эти движения. Первый столбец содержит номер, чтобы программное обеспечение могло идентифицировать состояние. Во втором описано движение от предыдущего положения к следующему. Последние два столбца описывают необходимое действие мотора С, чтобы выполнить движение из второй колонки. В таблицу включены некоторые тривиальные движения, такие, как перемещение из центра к центру. В таких случаях ничего не происходит, эти движения заданы на тот случай, если необходимо переместить робота в то положение, в котором он уже находится.

Таблица 3.1 – Случаи работы серводвигателей

ID движения	Движение	Число вращений	Направление вращения
0	Из центрального положения в центральное	нет	нет
1	Из правого положения в центральное	0,75	↑
2	Из левого положения в центральное	0,75	↓
3	Из центрального положения в правое	0,75	↓
4	Из правого положения в правое	нет	нет
5	Из левого положения в правое	1,5	↓
6	Из центрального положения в левое	0,75	↑
7	Из правого положения в левое	1,5	↑
8	Из левого положения в левое	нет	нет

Наклон может производиться из трех базовых положений: слева, справа и из центра. Это так называемые предыдущие положения. Также существуют 3 возможных движения: возвращение в центр, наклон вправо

и влево. В таблице 3.2 каждые из трех возможных и предыдущих движений ассоциированы с уникальным идентификатором номера.

Таблица 3.2 – Связь предыдущих и возможных движений с переменной

ID №	Движение	Предыдущая позиция
0	Возвращение в центральное положение	Центр
1	Наклон робота в правое положение	Правый наклон
2	Наклон робота в левое положение	Левый наклон

Таким образом с помощью простой формулы мы можем определить идентификационный номер состояний из первой таблицы:

$$ID = 3 * ND + PP, \quad (1)$$

Где ID - идентификационный номер состояний из первой таблицы,

ND – движение,

PP – предыдущая позиция.

Например, если предыдущая позиция это центр и нам необходимо наклонить робота влево:

$$ID = 3 * 2 + 0 = 6$$

Как и ожидалось, в таблице 3.1 под номером 6 находится переход из центрального положения в левое.

Ниже приведен код программы, осуществляющий наклон робота. Программа использует переменную «Tilt_Previous», чтобы сохранять предыдущую позицию наклона робота. Переключатель switch выполняет действия согласно таблице 3.1. Подпрограмма имеет единственный входной параметр: "Tilt_Current" – текущее движение. Таким образом, для работы подпрограммы необходимо знать только в какое положение нужно переместить робота – в правое, левое или центральное. На рисунке 3.1 приведен код подпрограммы.

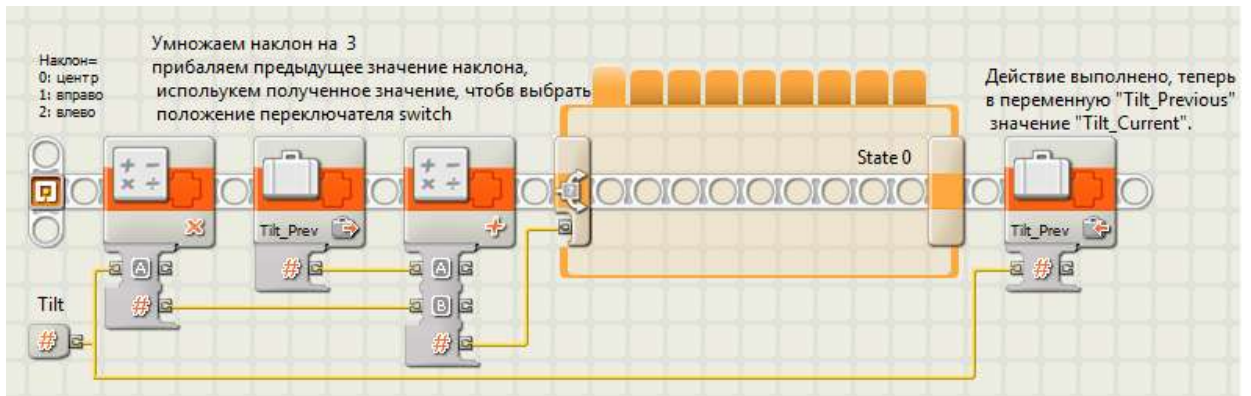


Рисунок 3.1 – Код подпрограммы, осуществляющий наклон робота.

На рисунке 3.2. приведены все вкладки блока switch.



Рисунок 3.2 – Вкладки блока switch.

ДВИЖЕНИЕ РОБОТА ВПЕРЕД И НАЗАД

Двигатель, подключенный в разъем В отвечает за движение ноги робота вперед и назад. Разработка подпрограммы, реализующей ходьбу, похожа на разработку предыдущей подпрограммы. Как и наклоны из стороны в сторону, движение ног робота вперед и назад может быть разбито на 9 отдельных случаев, показанных в таблице 3.3. В первой колонке находятся идентификатор случая, используемый подпрограммой. Во второй описывается движение ноги робота. Следует отметить, что фраза в колонке движение описывает какая нога (первое слово) в какое положение (второе слово) передвинется. В последних двух колонках описываются действия мотора В, необходимые для совершения движения из второй колонки.

Таблица 3.3 – Случаи работы серводвигателей

ID движения	Движение	Число вращений	Направление вращения
0	Центр в центр	нет	нет
1	Правая в центральную позицию	0,75	↓
2	Левая в центральную позицию	0,75	↑
3	Из центрального положения правая нога вперед, левая назад	0,75	↑
4	Правая вправо	нет	нет
5	Левая из переднего положения назад	1,5	↑
6	Из центрального положения левая нога вперед, правая назад	0,75	↓
7	Правая из переднего положения назад	1,5	↓
8	Левая влево	нет	нет

По аналогии с подпрограммой наклона, подпрограмма ходьбы использует единственную переменную, содержащую предыдущие

состояние шага: левая нога спереди, правая нога спереди, или обе ноги в центральной позиции. Таблица 3.4 три возможных движения при ходьбе и 3 возможных положения ассоциируются с уникальным номером.

Таблица 3.4 – Связь предыдущих и возможных движений с переменной

ID №	Движение	Предыдущая позиция
0	Обе ноги в центральное положение	Обе ноги в центральном положении
1	Правая нога вперед, левая назад	Правая нога спереди, левая сзади
2	Левая нога вперед, правая назад	Левая нога спереди, правая сзади

По аналогии с подпрограммой наклона можно использовать простую формулу для определения состояния в таблице 3.3.

$$ID = 3 * ND + PP, \quad (2)$$

Где ID - идентификационный номер состояний из первой таблицы,

ND – новое движение

PP – предыдущая позиция.

Код подпрограммы ходьбы очень похож на подпрограмму наклонов. Для того, чтобы сохранить предыдущую позицию шага используется переменная "Stride_Previous". Переключатель switch выполняет действия согласно таблице 3.3. У подпрограммы только один входной параметр номер ID из таблицы 3.4. Таким образом, при использовании этой подпрограммы необходимо знать только какое действие следует сделать роботу: обе ноги по центру, левую ногу вперед или правую ногу вперед.

На рисунке 3.3 приведен программный код подпрограммы движения.

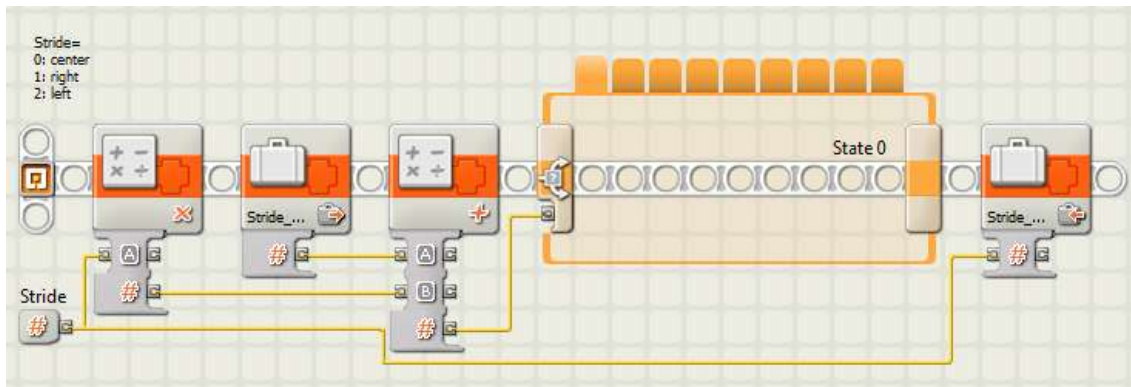


Рисунок 3.3 – Программный код подпрограммы движения

На рисунке 3.2. приведены все вкладки блока switch.



Рисунок 3.4 – Вкладки блока switch

СЛОЖНЫЕ ДВИЖЕНИЯ

Ходьба вперед и назад, повороты влево и вправо являются сложными движениями и могут быть достигнуты путем комбинации простых движений шага и наклона в нужной последовательности. Например, чтобы робот пошел вперед, необходимо выполнить следующие действия: наклон влево, движение правой ноги вперед, наклон вправо, движение левой ноги вперед. Исходный код, выполняющий движение робота вперед представлен на рисунке 3.5.



Рисунок 3.5 – Исходный код, выполняющий движение робота вперед

Подпрограмма, осуществляющая поворот робота представлена на рисунке 3.6.



Рисунок 3.6 – Подпрограмма, осуществляющая поворот робота

СКАНИРОВАНИЕ ПРЕПЯТСТВИЙ

Двигатель А (подключены в разъем А) предназначен для одновременного движения рук вверх и вниз и для движения ультразвукового датчика. Подпрограмма, приведенная на рисунке 3.7, сначала поворачивает ультразвуковой датчик влево и снимает с него показания, а затем поворачивает его вправо и также снимает показания. Эти показания хранятся в двух переменных, одна хранит показания слева, другая справа. Основная программа использует эти показания для того, чтобы определиться что делать, если ультразвуковой датчик определяет препятствие впереди робота. Между движением двигателя А и снятием показаний вставлены паузы. Они необходимы для стабилизации ультразвукового датчика после механического движения и снятия более точных показаний.

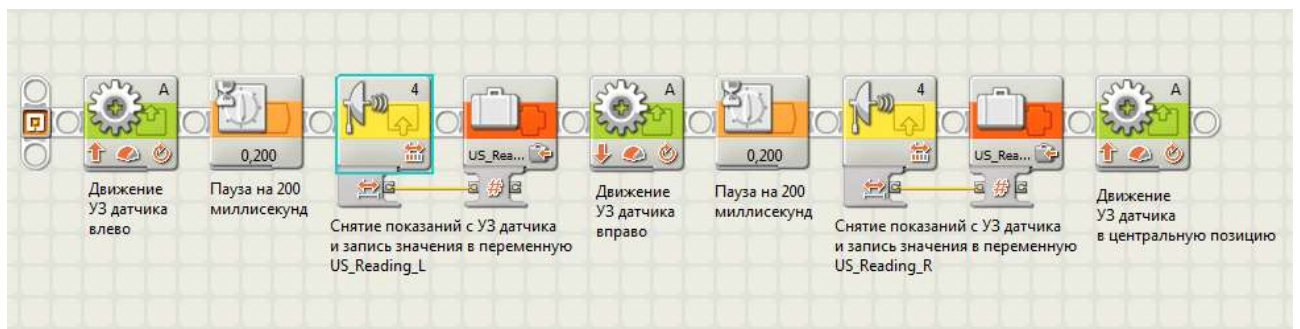


Рисунок 3.7 – Подпрограмма, реализующая движение ультразвукового датчика и снятие с него данных

ПОЛНЫЙ ПРОГРАММНЫЙ КОД

Программный код, показанный ниже, связывает все рассмотренные подпрограммы в конечную программу движения робота с огибанием препятствия. Программа начинается с выполнения подпрограммы сканирования. В ходе ее выполнения в переменные `US_Reading_L` и `US_Reading_R` записывается информация о расстоянии до объектов слева и справа. Далее делается проверка, есть ли препятствия слева или справа на

расстоянии меньше чем 15 дюймов. Если есть, то определяется с какой стороны оно находится и производится поворот в другую сторону. Если нет, то выполняется подпрограмма движения робота прямо.

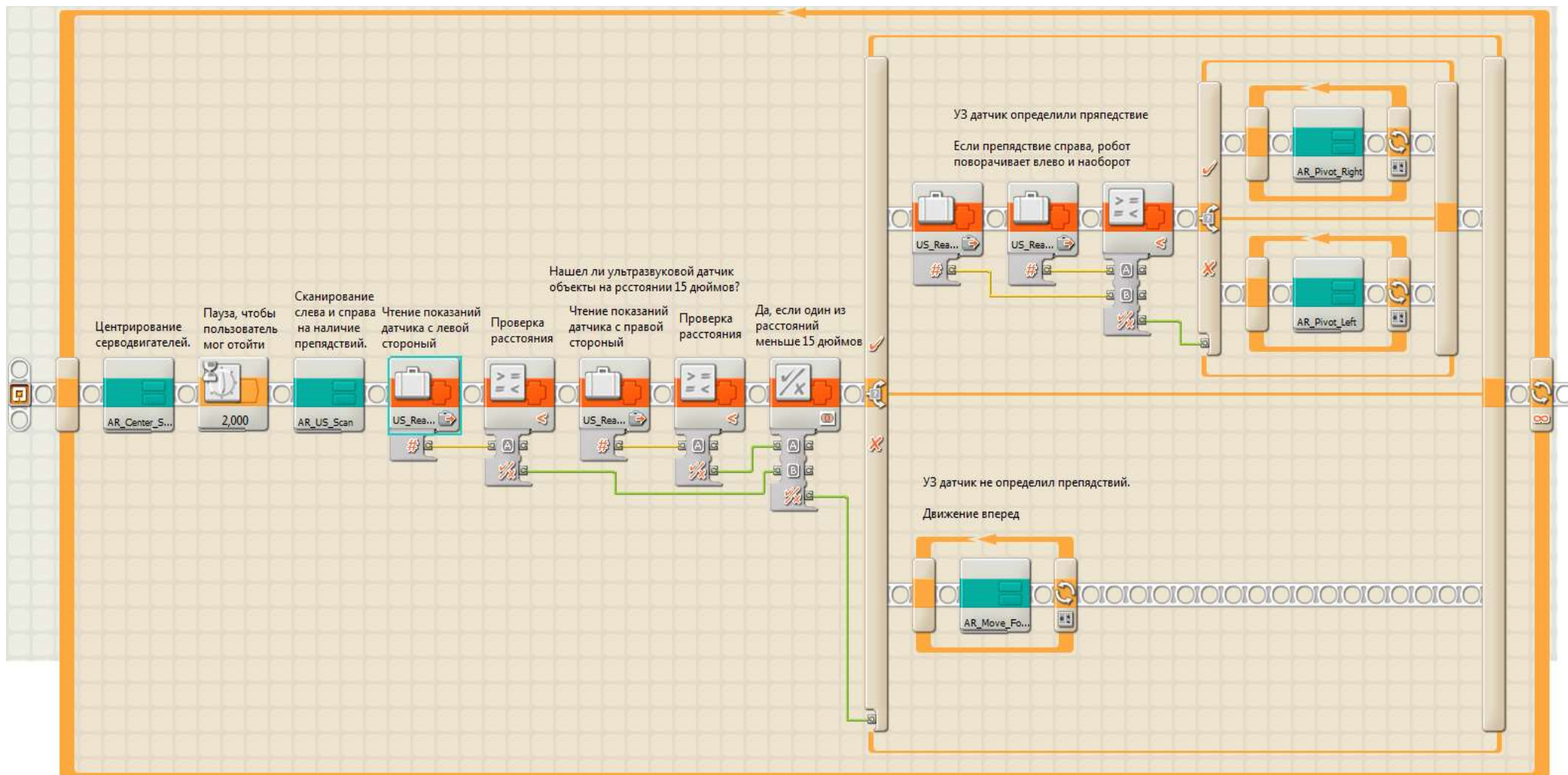


Рисунок 3.7 – Программный код, реализующий движение робота с огибанием препятствия.

