

Санкт-Петербургский государственный университет
Математико-механический факультет

Среда программирования роботов

QReal:Robots

Курсовая работа студента 245 группы
Пышновой Александры Витальевны

Научный руководитель

ст.преп. Ю.В.Литвинов

Санкт-Петербург
2012

Оглавление

Введение.....	3
1.QReal:Robot.....	2
2.Обзор существующих сред программирования.....	4
3.Цели.....	5
4.Реализация.....	5
Итоги.....	9
Список литературы.....	10

Введение

С каждым годом технологии развиваются активнее. В наше время появилось большое число контроллеров и автоматических систем управления, в качестве которых используются встраиваемые системы. Обычно такие системы состоят из микроконтроллера и дополнительных интерфейсов управления оборудованием.

На данный момент существует множество роботизированных конструкторов, предоставляющих возможность изучения процесса программирования микроконтроллеров. Примером такого конструктора может служить Lego Mindstorms, получивший популярность благодаря простоте использования и широкой доступности.

В настоящее время обучение программированию роботов активно внедряется в школы. Изучение кибернетики вызывает интерес как и из-за первоначального этапа сборки робота, так и из-за возможности написания программы и возможности наблюдения за тем, как эту программу выполняет реальный объект. Желание облегчить программирование задач вызвало необходимость создания функциональной визуальной среды разработки с дружественным интерфейсом, способной работать с такими конструкторами как Lego Mindstorms. Такие программы позволяют вырабатывать новые подходы и внедрять современные технологические средства в обучение.

Создано большое количество сред программирования роботов, одной из которых является QReal:Robots, являющаяся свободным средством и разрабатываемая силами преподавателей и студентов кафедры системного программирования. Целью этой курсовой стало добавление новой функциональности в QReal:Robots для выполнения реальных задач, в частности тех, что возникли при подготовке к городским и всероссийским соревнованиям по робототехнике.

1.QReal:Robots

QReal:Robots - среда обучения основам программирования и кибернетики, базированная на кроссплатформенном инструменте QReal, предназначенном для создания специализированных средств визуального программирования. Целями проекта QReal:Robots можно назвать избавление от недостатков, присущих большинству существующих инструментальных средств в области программирования роботов, и разработка удобного и дружелюбного интерфейса, поскольку большей частью пользователей таких сред являются школьники.

2.Обзор существующих сред программирования.

На данный момент существует много языков и сред программирования роботов. Множество этих инструментальных средств делятся на две группы: графические (NXT-G, Robolab, LabVIEW) и текстовые, основанные на существующих языках программирования (RobotC, leJOS, NXC). Самыми распространенными являются NXT-G, Robolab и RobotC.

1) NXT-G - графическая среда программирования, разработанная National Instruments для Lego, которая предоставляется в стандартном наборе конструктора. Написание программы с помощью NXT-G похоже на создание блок-схемы. Программист задает схему управления роботом, используя существующие блоки, описывающие характеристики его поведения. Данный способ создания программ является наглядным и подходит для людей с разными уровнями подготовки. Так же к преимуществам данной среды программирования можно отнести то, что она является case-редактором от Lego на базе LabView и может использовать LabView для создания "My Block". "My Block" - возможность создания собственного блока, который можно использовать в написании любой программы. Однако у NXT-G существует ряд недостатков, например, медленный редактор, недостаточная функциональность, возможность работы только с NXT.

2) Robolab - графический CASE-инструмент на базе LabView от компании National Instruments, наиболее часто используемый в школах и вузах для обучения кибернетике. В

отличие от NXT-G, Robolab позволяет создавать более сложные программы с нетривиальными математическими вычислениями и использовать как числа целого типа, так и числа с плавающей точкой. Так же данный инструмент имеет довольно продвинутый механизм логирования информации. Весомыми недостатками Robolab являются отсутствие пошаговой отладки и значительная с точки зрения российского потребителя стоимость.

3) RobotC - текстовая среда программирования, основанная на стандартном языке C. RobotC имеет два режима: для новичков (часть функциональности языка скрыта) и для людей, имеющих опыт в программировании. Одной из возможностей среды является управление роботами с помощью джойстика. Так же данный инструмент позволяет отлаживать программу и получать обновления показаний с датчиков в режиме реального времени. Текстовое представление можно отнести как к преимуществам, поскольку позволяет полностью контролировать написание кода, так и к недостаткам, поскольку оно является не столь наглядным как графическое представление и требует умения программировать на языке C.

3.Цели

Цели данной работы можно разделить на две части : исправление ошибок, таких как корректная заливка программы на робота по USB, правильное подключение к роботу при выборе модели, и введение новой функциональности, такой как создание нового блока “Ждать свет” и поддержка дополнительного аппаратного обеспечения.

4.Реализация

Существует возможность взаимодействия с роботом, используя внешние устройства (такие как компьютер, другой микроконтроллер или любое устройство, поддерживающее Bluetooth и использующее профиль SPP (Serial Port Profile)), осуществляемая с помощью прямых команд. Главная задача такого способа управления

заключается в создании простого интерфейса для использования всех ресурсов и функциональностей микроконтроллера без необходимости написания и запуска специальных программ для дистанционного управления. Команды делятся на несколько типов: системные и прямые, требующие ответа и не требующие после отправки пакета с указаниями. Поэтому роботом можно не только управлять с внешних устройств, а также можно обмениваться с ними информацией.

Поскольку прямые команды делают более простым взаимодействие внешних устройств со встроенными системами, они были использованы для реализации таких методов для управления роботом по USB, как SendI2C (для поддержки сенсора расстояния), и KeepAlive (для того, чтобы программа своевременно узнавала об отключении робота).

1. SendI2C

Сенсор расстояния является датчиком, входящим в стандартный набор Lego Mindstorms. С его помощью робот может не только различать перед собой препятствия и обходить их, но и определять точное расстояние до них, а также их габариты. Принцип работы данного устройства - измерение расстояния путем расчета времени, которое потребовалось звуковой волне для возвращения после отражения от объекта. Поэтому сенсор расстояния так же называют ультразвуковым датчиком. Передача информации между главным процессором и такими сенсорами, как сенсор звука, света и касания, а так же сервомоторами происходит через сопроцессор, а между процессором и сопроцессором обмен данных происходит по шине I2C. У ультразвукового датчика имеется собственная микросхема обработки информации, поэтому он связывается напрямую с главным процессором через другой канал I2C. I2C (Inter-Integrated Circuit) - последовательная шина данных для связи интегральных схем, используемая для соединения низкоскоростных периферийных компонентов со встраиваемыми системами и мобильными телефонами.

Для поддержки ультразвукового сенсора в программе был реализован метод SendI2C с использованием нескольких прямых команд, как требующих ответа, так и не

требующих. Этот метод отличается от метода Send (так же используемого для получения отправки прямых команд и получения данных с робота) тем, что взаимодействие происходит непосредственно с шиной I2C, а у Send - с сопроцессором. Send I2C приходит в действие во время исполнения программы, включающей в себя блок "Ждать расстояние". В SendI2C формируются пакеты с командами, отправляются на робота и принимается информация с показаниями с датчика. Метод вызывается до тех пор, пока полученные данные не будут удовлетворять поставленному условию в блоке.

2.KeepAlive

В панели управления QReal:Robots имеется кнопка, показывающая в каком состоянии находится сейчас робот : в подключенном или отключенном. Для того, чтобы программа узнавала об отключении робота и сразу же отображала изменение, был реализован метод KeepAlive, написанный с использованием прямой команды, требующей ответа, с тем же названием. Принцип работы заключается в том, что формируется пакет с указаниями, отправляется роботу и в ответ принимается другой пакет данных. В зависимости от того, принят ответ или нет, делается вывод о состоянии робота. Для того, чтобы отображение изменений происходило своевременно, команда KeepAlive посылается каждую секунду.

Создание нового блока "Ждать свет".

Необходимость создания нового блока "Ждать свет" появилась во время городских соревнований по робототехнике, а именно из-за участия в категории "Движение по линии". В этом задании роботу надо двигаться по черной линии на белом поле и уметь распознавать цвет, над которым в данный момент времени он находится. В QReal:Robots уже были реализованы блоки "Ждать цвет" и "Интенсивность цвета", но проблема заключалась в том, что для участия в соревнованиях необходимо загружать программу непосредственно на робота с установленной операционной системой, а QReal:Robots взаимодействует с операционной системой NXT OSEK, которая не поддерживает датчики распознавания цвета. Датчики света не умеют различать цвета или выделять определенный, но благодаря возможности измерения освещенности можно

определить над какой поверхностью находится робот: над темной или светлой. Таким образом возникла задача по созданию нового блока.

Для реализации поставленной задачи вначале был добавлен сам блок “Ждать свет” с такими полями, как “Порт” (для выбора порта, к которому подключен сенсор света), “Проценты” (означает необходимую степень освещенности), “Считанное значение” (можно выбрать при каком соотношении между этим полем и процентами завершится выполнение блока). Далее потребовалось создать класс `WaitForLightSensorBlock`, описывающий поведение программы во время выполнения блока, с методами начала работы, конца и условия, при котором будет работа будет продолжаться. Блок `WaitForLightSensor` взаимодействует с `RobotModel` (логическая модель), предоставляющая собой самого робота и его отдельные части (сенсоры, моторы). Также необходимо было создать класс `LightSensor`, который и выступает в роли датчика освещенности. О том, что происходит с показаниями с сенсоров, знает только блок. Он дает указания и ожидает ответ, полученные данные передают через логическую модель обратно блоку.

На рисунке 1. показан пример диаграммы с блоком “Ждать свет”.

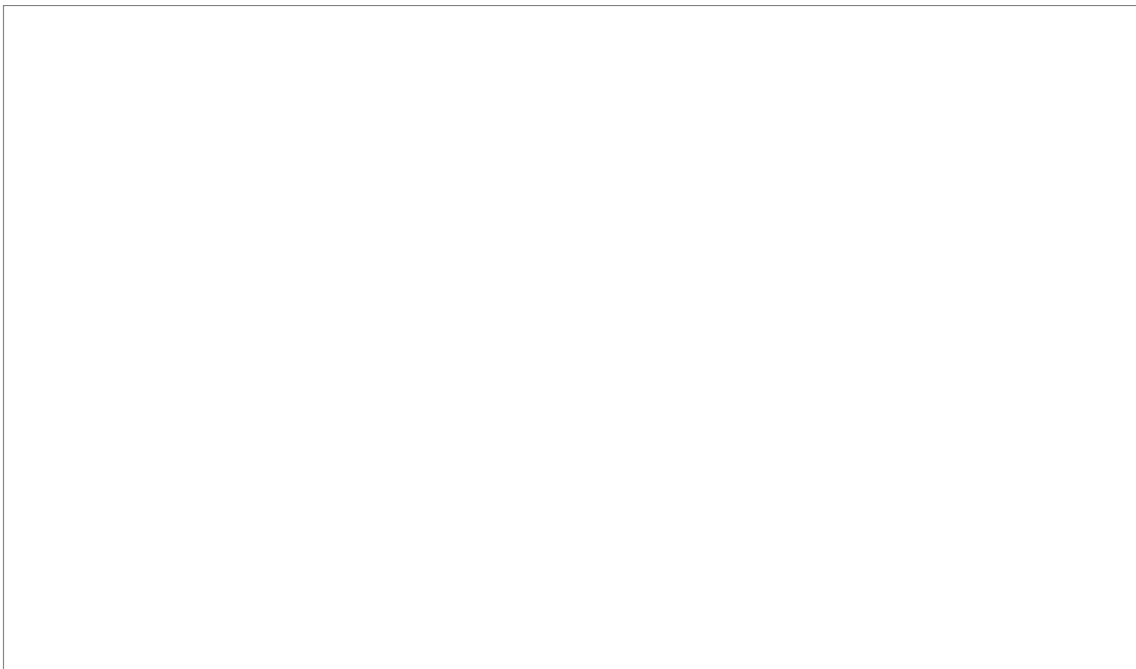


Рисунок 1.

Информацией о существовании блока располагает графическая модель, которая хранится в классе Interpreter, там же запускается выполнение диаграммы поведения робота. Для того, что программу можно было выполнить, необходимо выбрать модель робота (если модель реальная, так же выбирается способ подключения) и, в зависимости от того какая она, создается объект класса AbstractRobotModelImplementation. В этом классе создаются объекты типов SensorImplementations, MotorImplementations, BrickImplementations, отвечающие за взаимодействие с отдельными частями робота, и конфигурируются на указанных портах. Если выбранная модель является пустой или 2д-моделью, информация получается не с датчика, а с помощью специальных методов. Для работы блока “Ждать свет” для реальной модели был создан класс BluetoothLightSensorImplementation. В нем описано непосредственное взаимодействие с датчиком освещенности: то, какие команды ему нужно посылать, для получения показаний, и метод обработки полученных данных, так как они поступают не в том формате, в котором хотелось бы (в блоке нужно указывать проценты), и отправка их через модель обратно блоку. Отправка так же происходит, учитывая указанный способ подключения (Bluetooth, USB).

Итоги

В процессе работы над поставленными задачами было выполнено :

- проведен обзор существующих сред программирования роботов
- исправлен ряд ошибок в среде QReal:Robots
- добавлена новая функциональность
- произведена поддержка дополнительного аппаратного обеспечения
- произведена апробация на реальных задачах

Список литературы

1. Programming Solutions for the LEGO Mindstorms NXT. Which approach is best for you?
[Электронный ресурс] URL: <http://find.botmag.com/100701> (дата обращения : 29.05.2012)
2. Lego Mindstorms - Датчики.
[Электронный ресурс] URL: <http://mindstorms.ucoz.net/index/datchiki/0-7>
3. Lego Group, Appendix 2-LEGO MINDSTORMS NXT Direct commands // 2006
4. Р.М.Лучин, Программирование встроенных систем от модели к роботу // СПб, Наука, 2011 г, 184 стр
5. QReal // Сайт разработчиков URL: <http://www.qreal.ru/> (дата обращения : 29.05.2012)