

Системы управления знаниями. Знания организации есть обобщенная, систематизированная и прошедшая принятую в организации общественную экспертизу и относящуюся к существенным для деятельности организации аспектам информация. Для создания, распределения, хранения и привлечения знаний помогают системы управления знаниями [5].

Многообразие систем поддержки принятия стратегических решений позволяет применять различные методы к конкретным ситуациям и получать наиболее качественный результат, учитывая все возможные критерии и факторы, принять правильное стратегическое решение для дальнейшего улучшения и развития организации или предприятия.

Список литературы:

1. СППР – это... Что такое СППР? // Словари и энциклопедия на академике [Электронный ресурс]. – Режим доступа: <https://dic.academic.ru/dic.nsf/ruwiki/1133227>
2. Система поддержки принятия решений: помощник руководителя для стратегического и оперативного управления // ЕРАМ | Разработка ПО [Электронный ресурс]. – Режим доступа: <https://www.epam-group.ru/about/news-and-events/in-the-news/2009/sistema-podderzhki-prinyatiya-resheniy-pomoschnik-rukovoditelya-dlya-strategicheskogo-i-operativnogo-upravleniya>
3. Системы поддержки принятия решений // Боровское исследовательское учреждение [Электронный ресурс]. – Режим доступа: <http://bourabai.kz/tpoi/dss.htm>
4. А.А. Захарова Модели и программное обеспечение поддержки принятия стратегических решений в социально-экономических системах на основе экспертных знаний.
5. Д.М. Ершов, Р.М. Качалов Системы поддержки принятия решений в процедурах формирования комплексной стратегии предприятия / Предпринт # WP/2013/299. – М: ЦЭМИ РАН, 2013. – 60 с. (Рус.)

АНАЛИЗ ВОЗМОЖНОСТИ ИСПОЛЬЗОВАНИЯ ДАТЧИКА ОРИЕНТАЦИИ BOSH BNO055 ДЛЯ ОРИЕНТАЦИИ РОБОТА В ПРОСТРАНСТВЕ

*А.Д. Веретенников студенты группы 17В60, научный руководитель: Момот М.В.
Юргинский технологический институт (филиал) Национального исследовательского
Томского политехнического университета
652055, Кемеровская обл., г. Юрга, ул. Ленинградская, 26
E-mail: momotmvu@yandex.ru*

Цель работы получить сведения о возможности применения датчика BOSH BNO055 для пространственного ориентирования робота.

Современные датчики ориентации, применяемые в большинстве малобюджетных устройств, представляют собой электронные аналоги своих классических прототипов, но работают иначе. Например, электронный гироскоп не может показать положение устройства в пространстве, он передает угловую скорость по осям, а уже по ней следует вести непрерывный расчет поворотов устройства, что и позволяет следить за ориентацией. С электронным компасом ситуация также неоднозначная, так как у него нет встроенной стрелки с направлением на магнитный север. Он, конечно, передает напряженности магнитного поля по осям, но расчет по ним «вектора направления», скорее всего, покажет, что полюсов несколько, и направлены электронные стрелки совсем не на север и два разных прибора дадут различные результаты. Тоже будет и с акселерометром: электронный аналог акселерометра (на примере MPU-6050) очень сильно «шумит», и его показания нужно подвергать математической обработке, в противном случае зашкаливающие ускорения просто невозможно будет использовать.

Подключив к своему роботу гироскоп, акселерометр, магнитометр (электронный компас), следует еще применить значительный объем математических расчетов, чтобы получить из исходных данных значение ориентации робота. Будет правильно, если этим занимается отдельный процессор, который не отвлекается ни на что другое и постоянно принимает информацию от датчиков ориентации, обрабатывает её и фиксирует результат.

При работе с механизмом балансировки двухколесного робота на основе ATmega328 время, затраченное контроллером на вычисления и работу с датчиком ориентации составляет 1.4мс, частота запросов к MPU-6050 – не чаще чем 1 раз в 4мс. При этом точно узнать вектор направленности робота не представляется возможным, и постоянно требовалось сэкономить на расчетах, чтобы найти процессорное время для обработки другой информации.

Датчик BNO055.

Описание комбинированного датчика BNO055.

В свободной продаже по сравнительно низкой стоимости появился датчик производства BOSH BNO055 (рис. 1). Этот датчик значительно отличается от своих предшественников как по цене, так и по функционалу. Дело в том, что в BNO-055 есть свой модуль вычислений – это 32-битное ядро ARM Cortex M0, что позволяет производить большинство расчетов непосредственно внутри BNO-055 и пользоваться уже готовыми результатами.

Ниже приведена внутренняя архитектура описываемого датчика и его взаимодействие с внешним процессором, информация взята из документации BOSH[1].

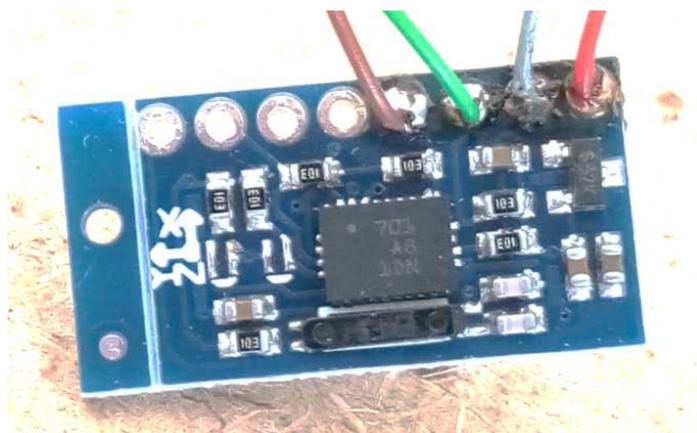


Рис. 1. Общий вид модуля BNO055

Ядро получает данные акселерометра, гироскопа и магнитометра, производит расчеты и сохраняет результаты. Результаты в дальнейшем могут быть запрошены внешними устройствами. В документации упоминается о том, что данный прибор можно использовать даже для навигации, т.е. его достаточно для нахождения не только углов поворота, но и для определения относительных координат и маршрута (инерционной навигации). Прибор может выдавать информацию по кватернионам, линейному ускорению, вектору гравитации, направлению движения устройства.

Согласно документации BNO055 может поддерживать три интерфейса обмена с внешними устройствами: I2C, HID-I2C, UART (Serial). Какой интерфейс будет применяться, задается логическими значениями на паре контактов микросхемы 5 (PS1) и 6 (PS0). Если оба контакта подтянуты к нулю, то с датчиком можно будет общаться по I2C протоколу. На следующей странице приведена стандартная схема подключения устройства при использовании его по I2C.

Области применения подобных устройств различны. В нашем случае BNO055 применяется для навигации мобильных роботов и квадрокоптеров.

Основные характеристики микросхемы Bosch Sensortec BNO055[1]:

- рабочие диапазоны акселерометра: ± 2 , ± 4 , ± 8 , $\pm 16g$;
- настраиваемая полоса ФНЧ акселерометра: 1 кГц – менее 8 Гц;
- рабочие диапазоны гироскопа: от ± 125 до ± 2000 °/с;
- настраиваемая полоса ФНЧ гироскопа: 523-12 Гц;
- рабочий диапазон магнитометра: ± 1300 мкТл по осям x, y, ± 2500 мкТл по оси z;
- разрешение магнитометра: ~ 0.3 мкТл;
- напряжение питания микросхемы: 2.4–3.6 В;
- диапазон рабочих температур: от -40 до 85 °С.

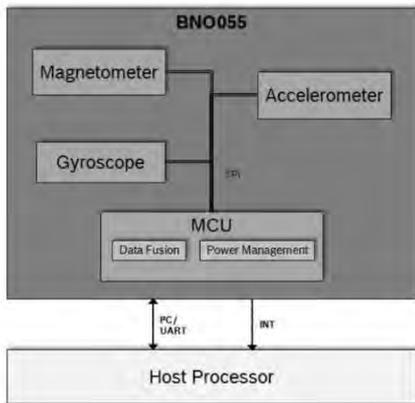


Рис. 2. Архитектура датчика BNO055

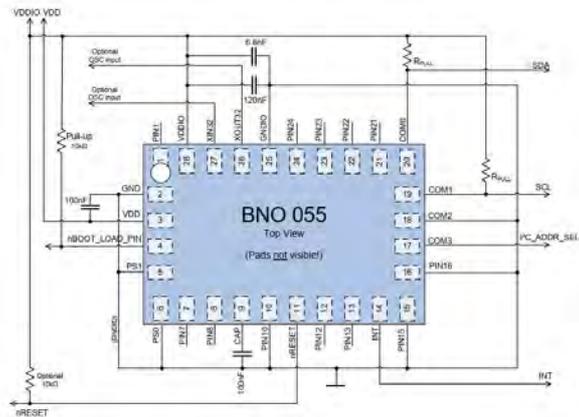


Рис. 3. Схема подключения BNO055

Осмотр модуля (рис. 1) выявил различия со всеми фото на сайтах продавцов, а именно: не были установлены (впяаны) перемычки (правее стрелок ориентации). Согласно расположению дорожек, этими перемычками задается интерфейс работы с внешними устройствами. При разомкнутых перемычках состояние было недокументированным, предполагалось, что при замкнутых (на землю) контроллер должен работать по I2C.

Собранная тестовая схема (рис. 4). Для создания схемы использовалась Arduino Nano. Питание платы также осуществлялось от 3.3В Arduino Nano. Nano была подключена к ноутбуку, от него и получала питание.

Сканирование I2C портов подтвердило предположение. До замыкания перемычек прибор не был виден.

Управление датчиком BNO055.

Следует воспользоваться ссылками, приведенными в [4] и установить поддержку датчика в среду Arduino IDE. Для управления регистрами BNO055 существует библиотека от Adafruit. Она с небольшими правками в подошла. Правки касались I2C адреса прибора, в библиотеке от Adafruit адрес был 0x28, а адрес имеющихся в наличии модулей – 0x29.

Для работы с датчиком BNO055 требуется следующее:

1. установить библиотеку Adafruit_BNO055-master в среду Arduino IDE;
2. установить библиотеку AdaSensor в среду Arduino IDE;
3. после установки библиотек станут доступны примеры использования BNO-055: rawdata, sensorapi, restore_offsets, bunny.

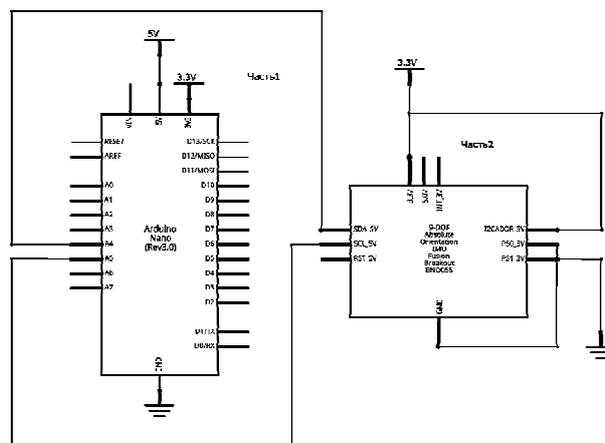


Рис. 4. Тестовая схема

1. rawdata позволяет отследить работу прибора при передаче им данных от отдельных датчиков и расчетных характеристики;.

2. `sensorapi` инкапсулирует и представляет информацию в виде классов.
3. `restore_offsets` позволяет восстанавливать информацию по калибровке из внешних источников. Это актуально, поскольку BNO055 не имеет своей энергонезависимой памяти и все данные по калибровке прибора будут утеряны после подключения питания. Сама же калибровка занимает довольно продолжительное время и до ее завершения данные будут искажены.
4. `bunny` позволяет визуализировать изменение положения прибора в виде образа кролика в среде `processing`.

Пример работы датчика BNO055.

Для данного примера потребуется установить среду программирования `processing` (загрузить и распаковать в какую-нибудь папку) <https://processing.org/download/>. После первого запуска папке «Документы» появится папка «Processing», а в ней папка «Libraries».

Скачать библиотеку Saito's OBJ Loader:

https://github.com/adafruit/Adafruit_BNO055/blob/master/OBJLoader/OBJLoader.zip

Скачать последнюю версию библиотеки GP4 GUI:

<https://sourceforge.net/projects/g4p/files/?source=navbar>

Распаковать полученные архивы в папку «Libraries».

Распаковать `Adafruit_BNO055-master.zip` в отдельную папку, открыть пример `bunny`, поменять адрес прибора, как указано выше, сохранить себе в рабочую директорию и загрузить в Arduino. Если все прошло удачно, можно приступать к визуализации.

После запуска среды `processing` (`Processing.exe`) открывать скетч для `Processing`, расположенный внутри папки `bunny` (`processing/cuberotate/cuberotate.pde`).

После запуска на выполнение загруженного скетча можно наслаждаться вращением кролика при помощи BNO055. При этом самого прибора касаться нельзя: он очень чувствительный и чуть что зависает напрочь. Лучше всего BNO055 и Arduino расположить на одной подставке, которую и вращать. За провода между BNO055 и Arduino также держаться нельзя.

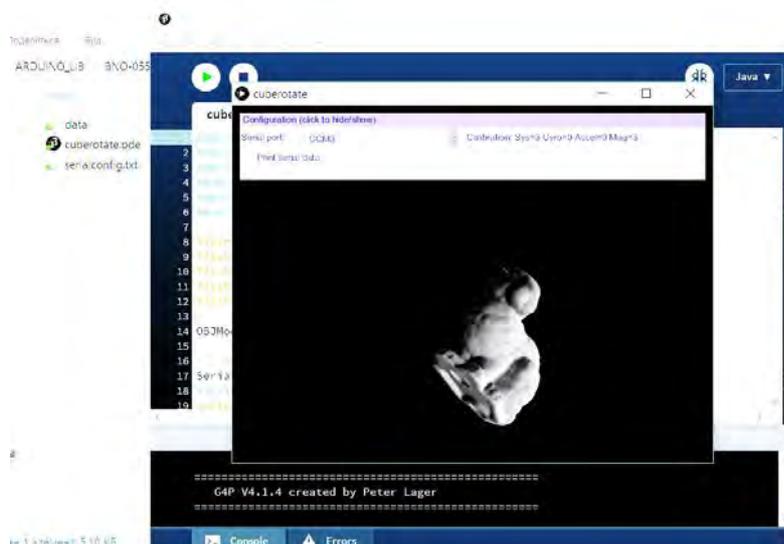


Рис. 5. Пример работы среды `Processing`

На системах с переключаемой графикой может возникнуть ошибка 1114. Ошибку можно устранить, выставив настройки энергосбережения на «Высокую производительность».

Список литературы:

1. BNO055: data sheet – № BST-BNO055-DS000-12. – дата выпуска – ноябрь 2014г. – 105 с.
2. BNO055: Handling, soldering & mounting instructions – № BST-BNO055-HS000-00. – дата выпуска – январь 2015г. – 24 с.
3. Townsend Kevin Adafruit BNO055 Absolute Orientation Sensor. – adafruit learning system. – 28 с.
4. Момот М.В. Датчик абсолютной ориентации BNO-055 <http://zizibot.ru/directory/sensor/bno055/>

5. Момот М. В. Мобильные роботы на базе Arduino. — СПб.: БХВ-Петербург, 2017. — 288 с.: ил. ISBN 978-5-9775-3741-4

РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ «АВТОЭКСПЕРТ» НА ОСНОВЕ ОНТОЛОГИИ «АВТОМОБИЛИ»

М.Б. Шабаш, студент магистратуры

*Научный руководитель: Авдеенко Т.В., д.т.н., профессор
Новосибирский государственный технический университет
630073, г.Новосибирск, пр-т К.Маркса, 20
E-mail: mayashabash@mail.ru*

Аннотация. В данной статье представлено описание построения онтологии «Автомобили» для разработки информационной системы «Автоэксперт». Данная система предназначена для людей, кто интересуется машинами и для помощи покупки той или иной машины. В рамках работы освещены различные точки зрения на понятие онтологии, используемого в современных информационных технологиях, дано определение этого термина, а также упоминаются возможные области применения онтологий в информационных системах. Рассмотрены преимущества редактора онтологий Protégé. Приводится описание методов построения онтологии предметной области. Детально описана последовательность действий. Показывается, что онтологии могут использоваться как при проектировании и разработке систем, основанных на знаниях, так и в качестве полноправного компонента во время функционирования системы. Описаны различные информационные запросы, которые может обрабатывать информационная система.

Ключевые слова: онтология, информационная система, предметная область, база знаний, классы
Введение

Существует множество определений понятия «онтология». Классическое определение онтологии, данное Грубертом в 1993 году, звучит так: «онтология – это эксплицитная спецификация концептуализации» Если смотреть более развернуто, то онтологией называют схему, состоящую из классов. Под определение онтологии попадают многие модели представления знаний: фреймы, семантические сети, концептуальные карты и т.п. Формальное определение онтологии можно записать в виде: $O = \{C, R, A\}$, где O – онтология, C – совокупность концептов (понятий) предметной области, R – совокупность отношений между концептами, A – набор аксиом (законов и правил, которые описывают законы и принципы существования концептов) [1]. В настоящее время, онтологии широко используются в обучении, различного рода исследованиях и информационных технологиях. Как своеобразная форма представления некоторой области знаний (предметной области) онтологии применяются при создании семантической паутины, систем искусственного интеллекта, а также прикладных информационных систем. Основной задачей в работе будет описание создания онтологии предметной области, которая будет являться базой знаний для информационной системы «Автоэксперт».

Основная часть

Обычно, онтология предметной области это не конечная цель реализации. Разработка онтологии нужна для набора данных, знаний, чтобы потом другие программы и приложения смогли воспользоваться ей. Среди таких приложений можно отметить современные интеллектуальные системы (ИС) – системы, способные решать задачи в некоторой предметной области, традиционно считающиеся творческими. Такие системы являются системами, основанными на знаниях (СОЗ), хотя иногда различие между этими двумя типами систем фактически не проводится. Основными компонентами ИС являются база знаний (декларативный компонент) и «решатель» или «механизм вывода» (процедурный компонент); в некоторых случаях в качестве важных составляющих ИС упоминаются также интеллектуальный интерфейс пользователя, подсистема объяснений и интеллектуальный редактор БЗ.

При создании онтологии следует иметь в виду, что не существует единственного «правильного» способа или методологии разработки онтологий – всегда существуют жизнеспособные альтернативы, однако содержание онтологии всегда определяется её целями (предполагаемым применением) и планируемыми направлениями дальнейшего развития (процесс создания онтологий, как правило, является итеративным). В общем виде онтология представляет собой набор элементов четырех типов: понятия (классы) – концептуальные сущности; экземпляры (индивиды) – конкретные представители понятий; отношения (предикаты) – связи между понятиями или экземплярами (таксономиче-