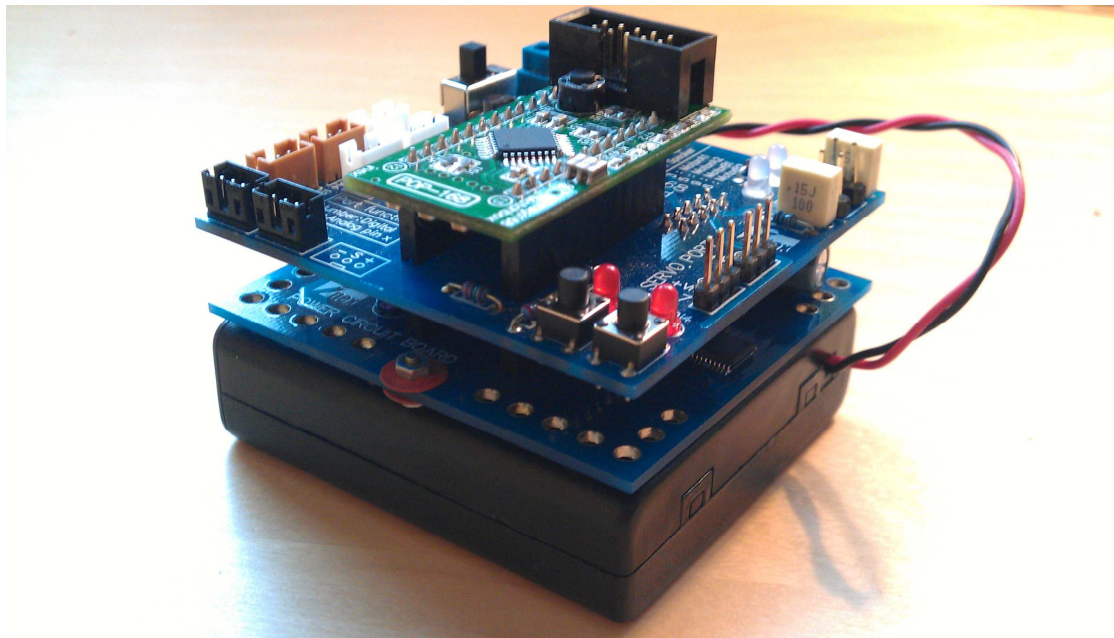


Управление роботизированной платформой на базе микроконтроллера ATmega168

Федоренко Ю.В.

*Московский государственный институт электроники и математики
Кафедра ИТАС*



Введение

В наши дни тема взаимодействия робота и человека становится все более актуальной. Многие предприятия и заводы едва ли можно представить без использования робототехники. Развивается наука, развивается и техника, следуя за прогрессом. Появляются мощные микроконтроллеры малых размеров и малыми ценами, что делает их доступными любому человеку.

В этой статье на примере управления роботизированной платформой будет показано как же все-таки происходит взаимодействие человека и программируемого контроллера.

Постановка задачи

В качестве роботизированной машины была выбрана конструкция, основанная на платформе с шасси, так как она может послужить неплохим примером для программирования контроллера и быстрым способом создания робота. К тому же, после овладения основными приемами обращения к микроконтроллеру, можно добавлять роботу дополнительную функциональность.

Существует множество микроконтроллеров и платформ, которые имеют схожую функциональность. Для выполнения данной задачи была выбрана плата на базе Arduino. Во-первых, это платформа, предназначенная для «physical computing» с открытым программным кодом. Во-вторых, её можно приобрести по относительно низкой стоимости, по сравнению с другими платформами. Например, самая недорогая версия модуля Arduino может быть собрана вручную, а многие готовые модули могут стоить меньше 50 долларов. В-третьих, среда разработки Arduino является кросс-платформенной, т.е. программное обеспечение Arduino работает под ОС Windows, Macintosh OSX и Linux. Также существует плагин для работы с Arduino в Microsoft

Visual Studio, который позволяет совместить проекты по управления платой Arduino с проектами, которые удобно писать под Visual Studio (например, обработка изображений, получаемых с камеры, с помощью библиотеки OpenCV).

Решение задачи

Платформа представляет из себя круглое основание, с двух сторон основания закреплены шаровые колеса (изготавливались на заказу по шаблону). Основание имеет в себе множество отверстий для удобства крепления модулей, деталей и моторов к ней.

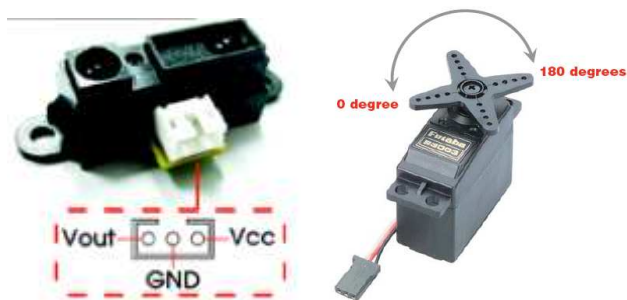


К этой платформе прикреплены два мотора-редуктора, которые вращают колеса, с рабочим напряжением от +3 до +9 постоянного тока и потреблением тока 130мА при напряжении +6В постоянного тока и отсутствии нагрузки со средней частотой вращения от 170 до 250 оборотов в минуту.

Также к платформе прикреплен стандартный RC-сервомотор, который позволяет осуществлять угловое перемещение в диапазоне от 0 до 180 градусов установленного на него инфракрасного датчика расстояния GP2D120 (по техническим характеристикам вращение до 180 градусов происходит за 1.5 секунды при максимальном рабочем напряжении 6В постоянного тока).

Датчик выполняет функцию инфракрасного измерителя расстояния, который иногда называют ИК-дальномером. С помощью этого датчика робот может определять расстояния и обнаруживать препятствия, используя инфракрасное излучение. Диапазон измеряемых расстояний лежит от 4 до 30 см при напряжении питания от 4.5 до 5 В, при потребляемом токе 33мА. Диапазон выходного сигнала от 0.4 до 2.4 В, при напряжении питания +5В.

Модуль соединяется с платой Arduino при помощи трехпроводного плоского кабеля, на концах которого разъемы с тремя выходами: вход напряжения питания ($V_{cc}=U_{\text{питания}}$), вывод общей шины (GND) и вывод выходного напряжения ($V_{out}=U_{\text{выход}}$).



Также к передвигающейся платформе прикреплен LCD-экран с последовательным управлением по 16 символов на 2 строки. Модуль требует одну линию ввода/вывода микроконтроллера, общую шину и напряжение +5В. Подключается к плате Arduino тремя контактами: напряжение питания +5В(+), последовательный вход данных (S) и общая шина(G).



Аппаратной частью робота служит платформа Arduino Nano построенная на микроконтроллере ATmega168. (ознакомиться с ее спецификацией можно на сайте <http://arduino.ru/Hardware/ArduinoBoardNano>).

Взаимодействие модулей и аппаратной части происходит через плату управления IO Expansion, на которой имеются: цифровые I/O-разъемы (для подключения, например, сервомотра), аналоговые разъемы (выводом питания и общей шины), разъемы для подачи напряжения питания, выходы для постоянного тока и, соответственно, разъем для модуля Arduino Nano.



Роботизированная платформа движется при помощи электродвигателей с редукторами, управление которыми осуществляется модулем контроллера Arduino Nano, программируемый через USB интерфейс посредством программного обеспечения Arduino. Язык программирования Arduino основан на C/C++. (ознакомиться с базовыми о данном языке программирования можно на сайте: <http://arduino.ru/Reference>)

Программа

Программа реализует движение робота. После запуска робота, он начинает движение вперед, если показания датчика имеют значения менее 500 (означает, перед ним нет препятствия). Датчик будет считывать данные, что нет объекта перед ним по ходу движения вперед. Как только датчик считывает значение, которое окажется больше или равно 500, роботу будет подан сигнал остановиться, чтобы найти другой

способ движения. Для этого он отъедет немного назад, остановится и запустит сервомотор, чтобы установленный на нем датчик вновь считал информацию о расстоянии. Все данные о расстоянии сравниваются между собой и робот поворачивается в направлении, где считанная информация имеет меньшее значение, то есть большее расстояние, после чего робот начинает то же самое движение, которое производил с самого начала. Все это повторяется до выключения питания.

Программа 2

Программа реализует движение робота, который должен найти объект, задействовав инфракрасный датчик. В нашем случае объектом служит мяч. После запуска робота, он начинает приводить в движение сервомотора, на котором установлен датчик, в поисках мяча. Сервомотор будет поворачиваться на 180 градусов и фиксировать показания дальномера через каждые 20 градусов. Затем эти показания сравниваются, и из них выбирается максимальное, что будет означать близость к объекту. В случае если все показания датчика окажутся меньше 20, робот развернется в противоположную сторону, т.к. это будет означать, что нет объектов в исследуемой области. Однако все показания датчика будут сравниваться с числом 280 (см. листинг программы). После сравнения показаний датчика робот направится в ту сторону, где показания датчика будут наиболее близкими к числу 280. Робот будет продолжать искать мяч, пока значения датчика не достигнут 280. Достигнув цели робот издаст звуковой сигнал, что будет означать завершение поставленной задачи.

Робот и компьютерное зрение

Наиболее интересной задачей в современном подходе к изучению роботизированных платформ является задача использования компьютерного зрения. Компьютерное зрение – это есть теории и технологии создания машин, которые могут производить обнаружение, слежение и классификацию объектов. «Компьютерное зрение относится к теории и технологии создания искусственных систем, которые получают информацию из изображений. Видеоданные могут быть представлены множеством форм, таких как видеопоследовательность, изображения с различных камер или трехмерными данными с медицинского сканера.»

Компьютерное зрение напрямую связано с машинным зрением, которое по сути является применением компьютерного зрения на практике. Например, обработка видеоданных для управления устройством, которая реализуется как программно, так и аппаратно.

Для нашей роботизированной платформы, основанной на Arduino, можно реализовать обработку видеоданных посредством установленной на платформе видеокамеры. Чтобы обрабатывать сигнал, полученный с видеокамеры, воспользуемся библиотекой по работе с компьютерным зрением OpenCV (**Open Source Computer Vision**)

OpenCV поддерживает C++, C, Python и Java интерфейсы на разных платформах (Windows, Linux, Android, Mac). Библиотека содержит в себе более 2500 различных оптимизированных алгоритмов и широко используется программистами во всем мире. На данный момент в процессе установки библиотеки OpenCV имеется ряд задокументированных проблем совместимости с различными версиями ОС и IDE. Поэтому до начала процесса установки необходимо внимательно ознакомиться с документацией.

Методы и подходы к распознаванию образов

В настоящее время существует множество алгоритмических разработок и программного обеспечения, реализующие предварительную обработку информации, автоматическую кластеризацию, планирование траекторий движения мобильных роботов в сложной среде, выполнение роботами интеллектуальных задач. При этом применяется интеллектуальное управление, генетические алгоритмы и другие методы робототехники, совмещенные с теорией искусственного интеллекта.

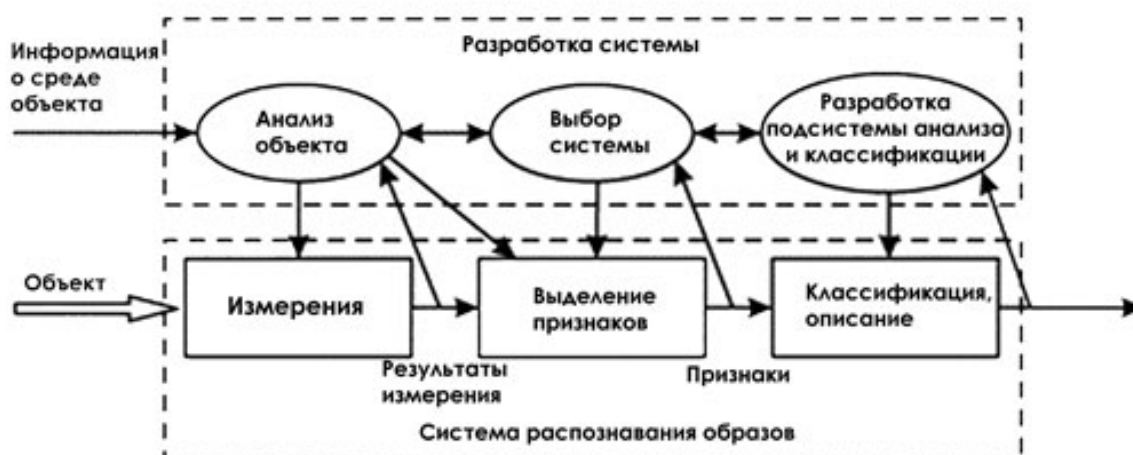
Также к методам обработки информации относятся распознавание образов и анализ сцен. Трудно назвать такую отрасль науки или сферу производственной деятельности, где распознающие системы не используются или не будут использоваться в ближайшее время. Распознавание образов -- это отнесение исходных данных к определенному классу с помощью выделения существенных признаков, выделяющих эти данные из общей массы, а также это процесс, которым можно управлять, изменяя поток поступающей информации или выбирая методы распознавания и их характеристики.

Можно выделить несколько методов распознавания образов. Самыми распространенными из них являются **методы перебора вида объекта, анализ характеристик образа, обучение нейронных сетей.**

Для **метода перебора** характерно сравнение объекта с базой всевозможных данных, где для каждого вида объектов представлены различные модификации отображения. Например, для оптического распознавания букв можно перебрать шрифт, свойства шрифта и т. д.

Для **метода анализ характеристик образа** характерны анализ свойств объекта, выделение контура объекта и исследовать его свойства (связность, наличие углов и другие)

Также существует метод, который основан на использовании искусственных нейронных сетей. Этот метод требует либо большого количества примеров задачи распознавания (с правильными ответами), либо специальной структуры нейронной сети, учитывающей специфику данной задачи. Отличают высокую эффективность и производительность этого метода для решения различных задач.



Выделяют несколько типов задач распознавания:

- задача распознавания – отнесение предъявленного объекта по его описанию к одному из заданных классов (обучение с учителем);

- задача автоматической классификации – разбиение множества объектов, ситуаций, явлений по их описаниям на систему непересекающихся классов (кластерный анализ, самообучение);
- задача выбора информативного набора признаков при распознавании. Качество распознавания во многом зависит от того, насколько удачно создан алфавит признаков, придуманный разработчиками системы. Поэтому признаки должны быть инвариантны к ориентации, размеру и вариациям формы объектов;
- задача приведения исходных данных к виду, удобному для распознавания;
- динамическое распознавание и динамическая классификация задачи 1 и 2 для динамических объектов;
- задача прогнозирования – суть предыдущий тип, в котором решение должно относиться к некоторому моменту в будущем.

При постановке задач распознавания образов чаще всего пользуются математическим описанием, стараясь заменить эксперимент логическими рассуждениями и математическими доказательствами. Отличие от теории искусственных нейронных сетей заключается в том, что в распознавании основой является получение результата с помощью математики, а не экспериментальным путем. Наиболее часто в задачах распознавания объектов рассматриваются монохромные изображения. Это делается для того, чтобы рассматривать изображение как функцию на плоскости. Если рассмотреть точечное множество на плоскости A , где функция $f(x,y)$ выражает в каждой точке изображения его характеристику – яркость, прозрачность, оптическую плотность, то такая функция есть формальная запись изображения. Множество же всех возможных функций $f(x,y)$ на плоскости A есть модель множества всех изображений X . Если ввести понятие сходства между образами, можно поставить задачу распознавания. Вид такой постановки зависит от последующих этапов при распознавании в соответствии с тем или иным подходом.

Описанные методы и задачи позволяют оценить возможность применения методики управления процессами распознавания образов и возможные эффекты от применения.

Использованная литература

Аркадьев А.Г., Браверман Э.М. Обучение машины классификации объектов - М.: Наука. 1971

Гонсалес Р., Вудс Р. Цифровая обработка изображений. - Техносфера, 2005

Ту Дж., Гонсалес Р. Принципы распознавания образов. – Мир, 1978

Дуда Р., Харт П. Распознавание образов и анализ сцен. – Мир, 1976

А. Фор Восприятие и распознавание образов. – Машиностроение, 1989