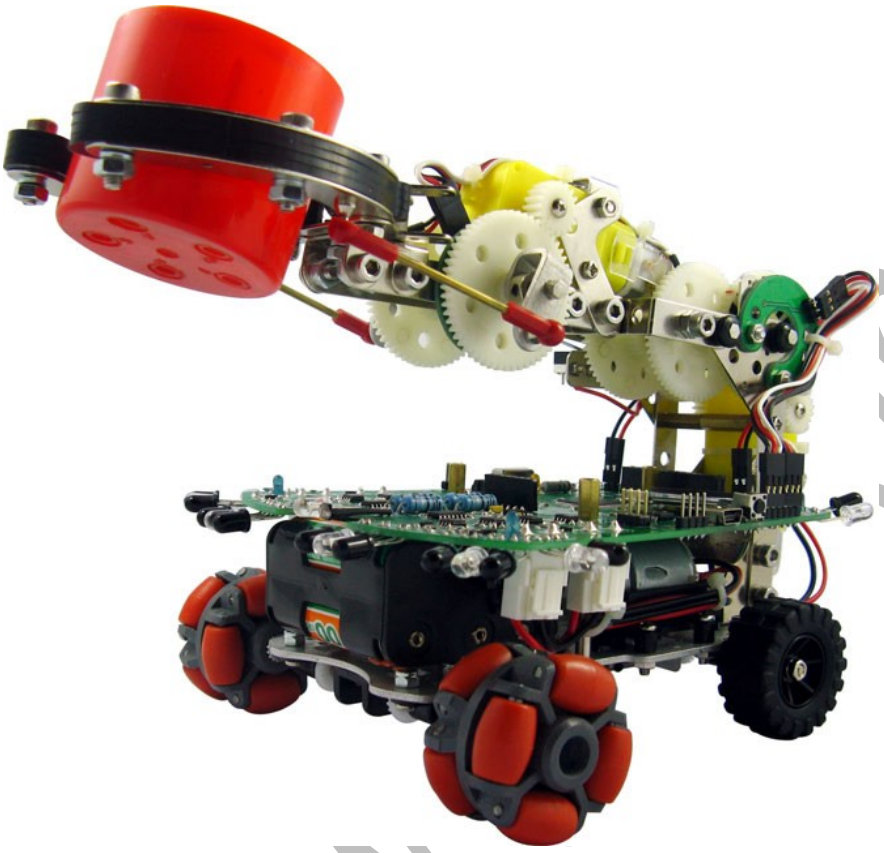
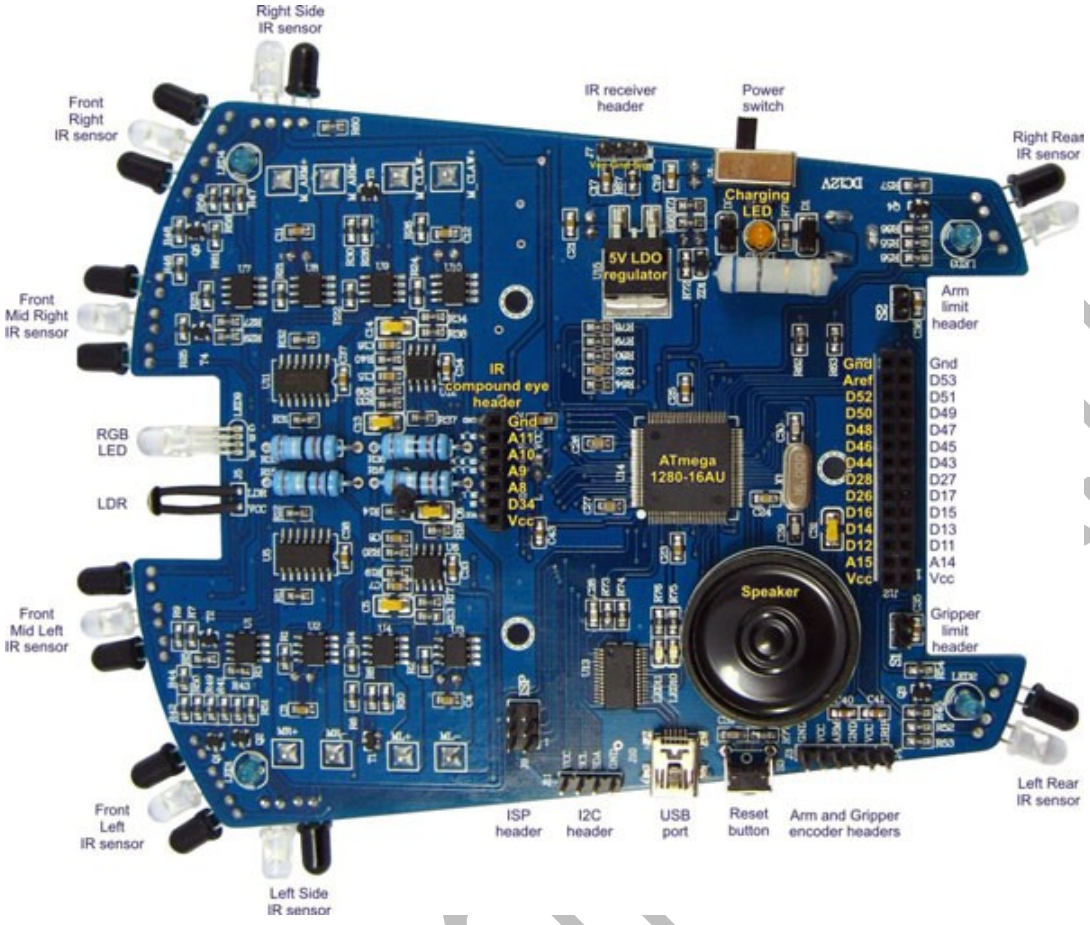
Mr.Tidy

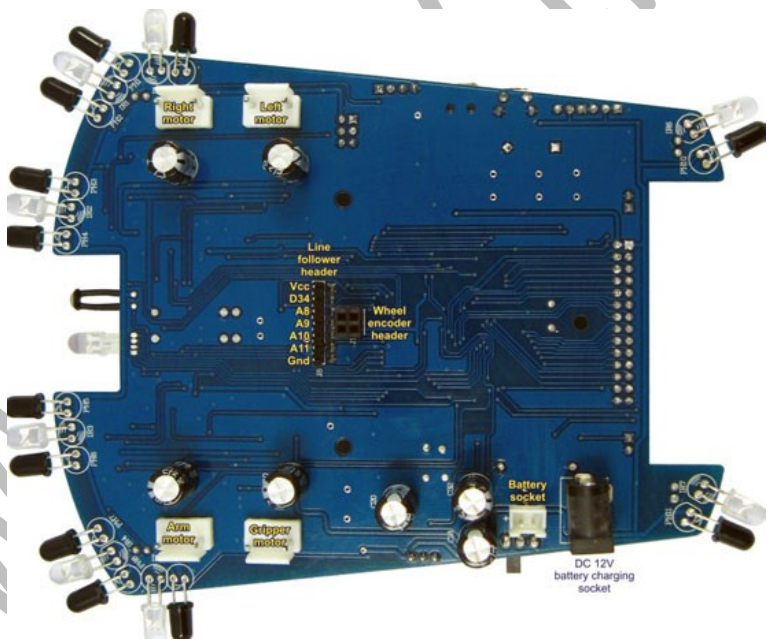


Мистер Тайди является уникальной многофункциональной роботизированной платформой, идеально подходящей как для студентов, так и для просто увлекающихся робототехникой людей любого возраста. Благодаря уникальному манипулятору и набору оптических сенсоров, робот способен решать огромное количество задач. Среда разработки является полностью открытой и позволяет очень легко программировать Мистера Тайди, а так же подключать к нему различные дополнительные устройства.

Содержание:

Основные элементы платы

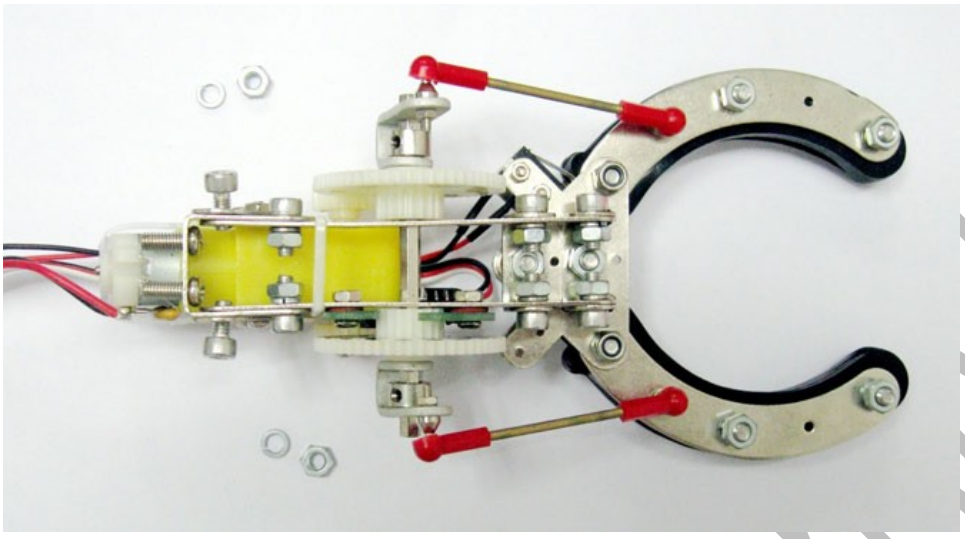




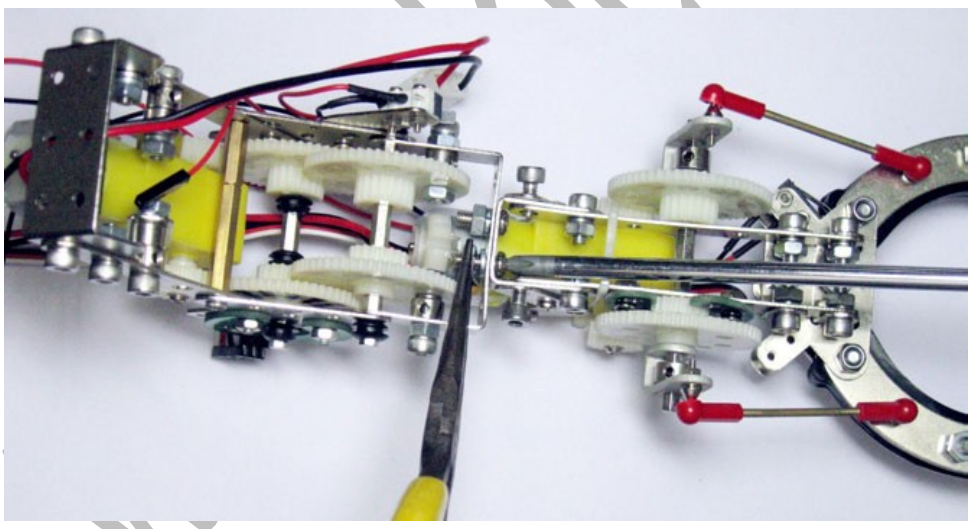
Сборка

ШАГ 1: Манипулятор робота состоит из 2 основных частей, которые должны быть предварительно соединены. Начните с раскручивания гаек, как показано на рисунке ниже. Винты полностью вынимать не следует. После этого, можно соединить обе части манипулятора.

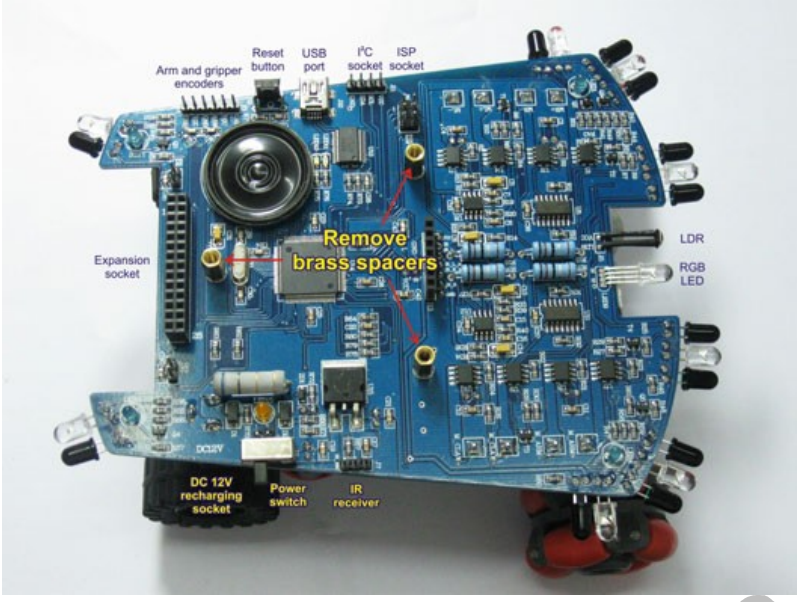
.



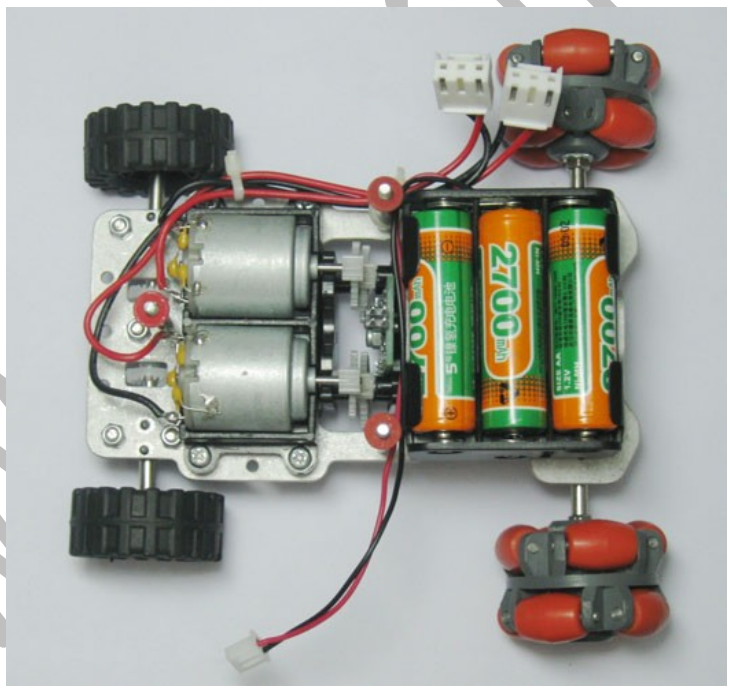
ШАГ 2: Соедините 2 части манипулятора, как показано ниже и закрутите их с помощью гаек и шайб, вынутых на первом шаге.



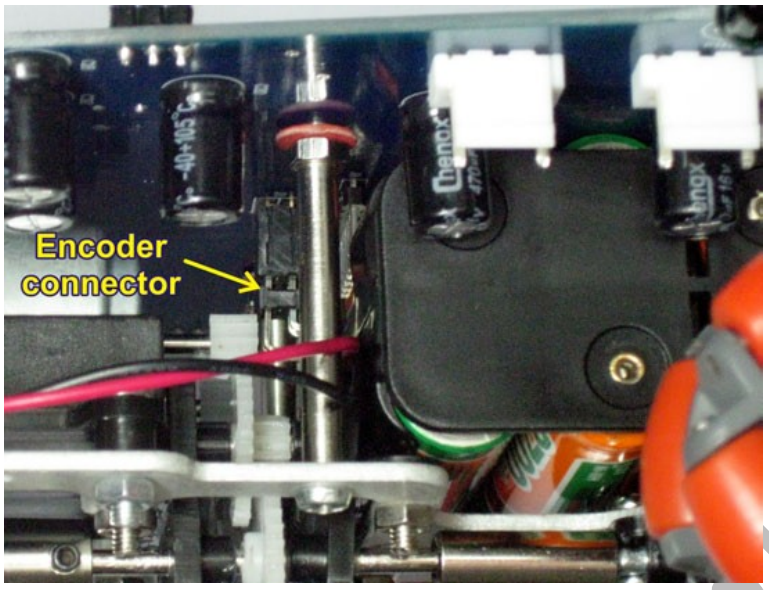
ШАГ 3: Раскрутите металлические стойки и удалите плату.



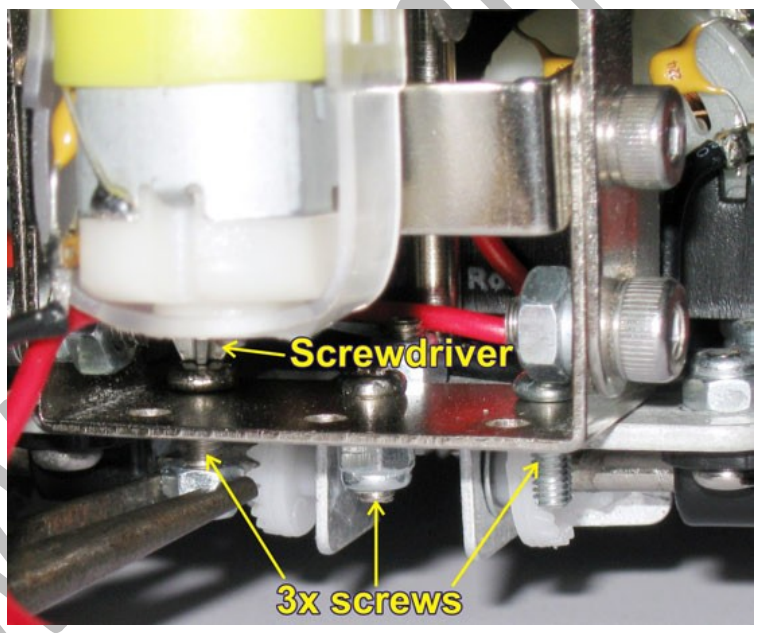
ШАГ 4: Вставьте 6 пальчиковых NiMh аккумуляторов типа “AA”. Не используйте алкалиновые батарейки, так как они не обладают достаточной энергией для питания робота, и они не будут перезаряжаться при подключении кабеля.



ШАГ 5: Установите плату заново. Обратите внимание, что бы датчик вращения колес был расположен так, как показано на рисунке и что бы провода питания не соприкасались с шестеренками.



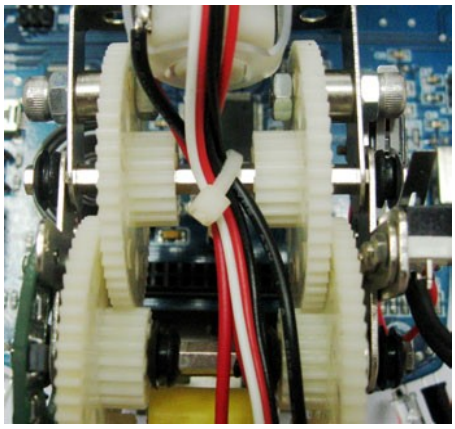
ШАГ 6: Поместите манипулятор на платформу и закрепите его с помощью 3x8мм винтов, как показано на рисунке. Рекомендуется использовать гаечный ключ или плоскогубцы

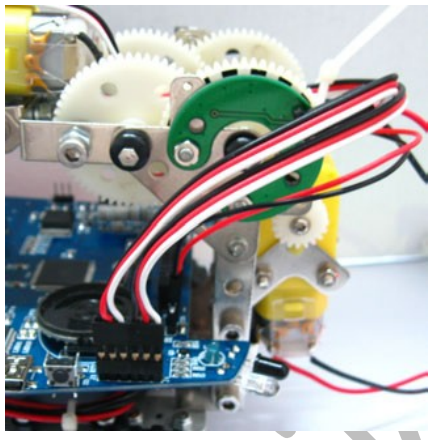


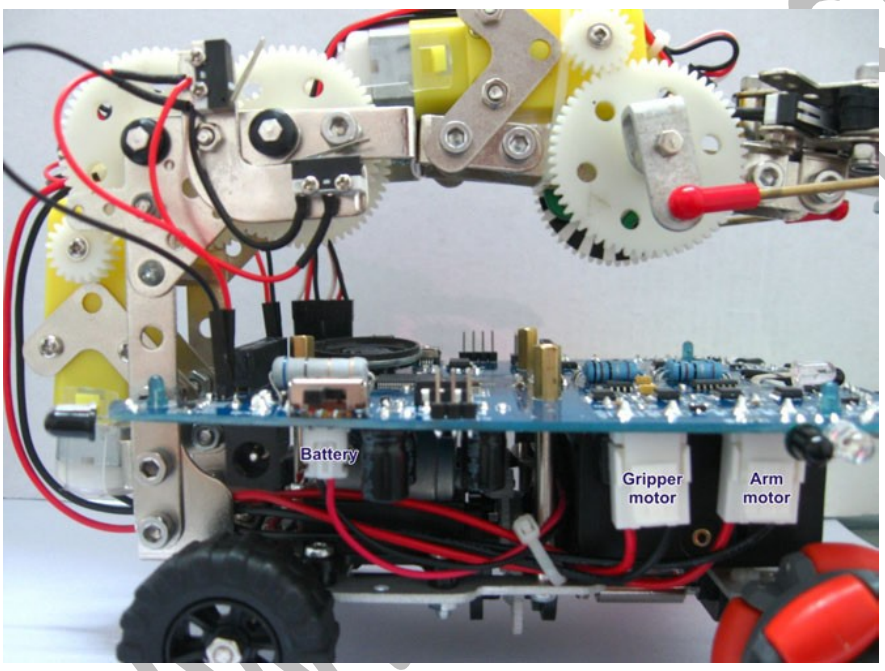
.

ШАГ 7: Теперь, когда манипулятор установлен, мы должны закрепить все провода на раме и подключить их к плате. Обратите внимание, что разъемы должны быть соединены именно так, как показано ниже.

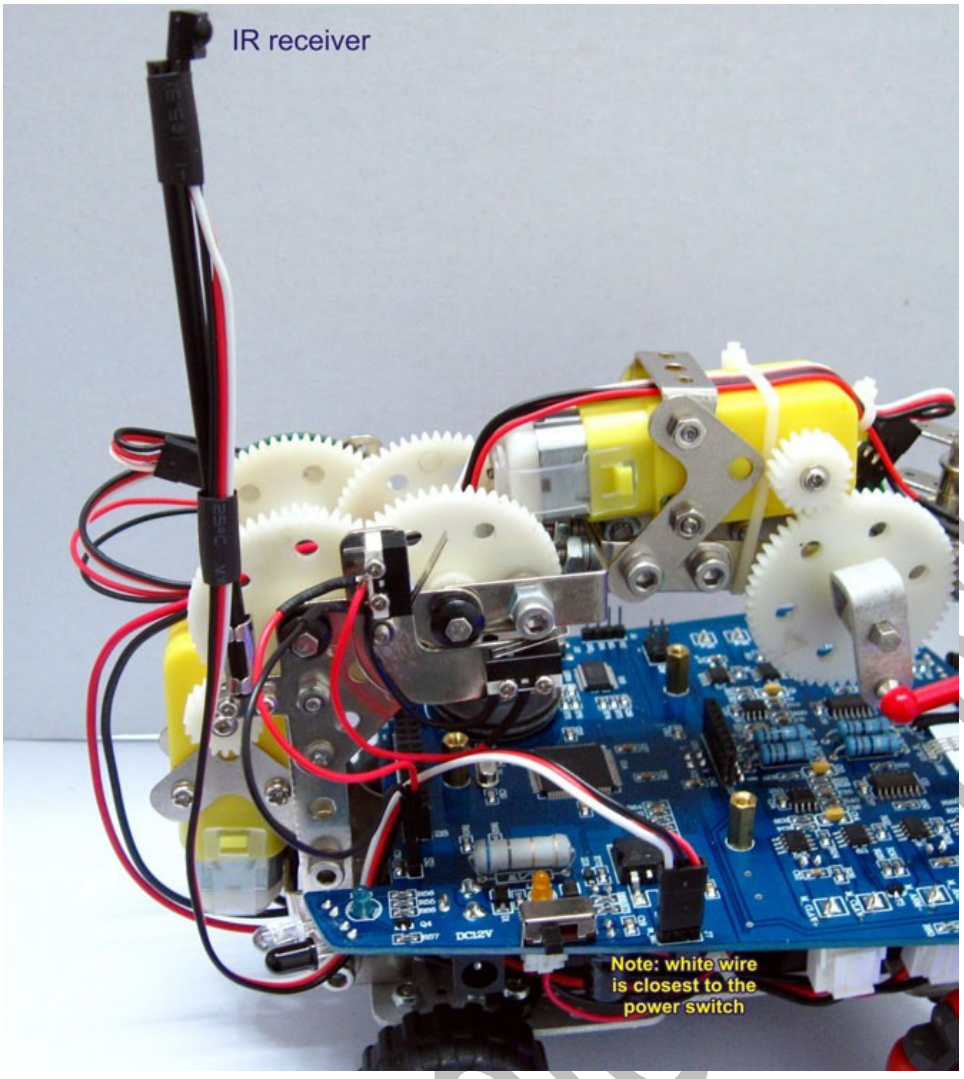
.







ШАГ 8: Установка инфракрасного приемника к роботу. Обратите внимание на порядок подключения проводов к микроконтроллеру. Неверное соединение может повредить приемник.

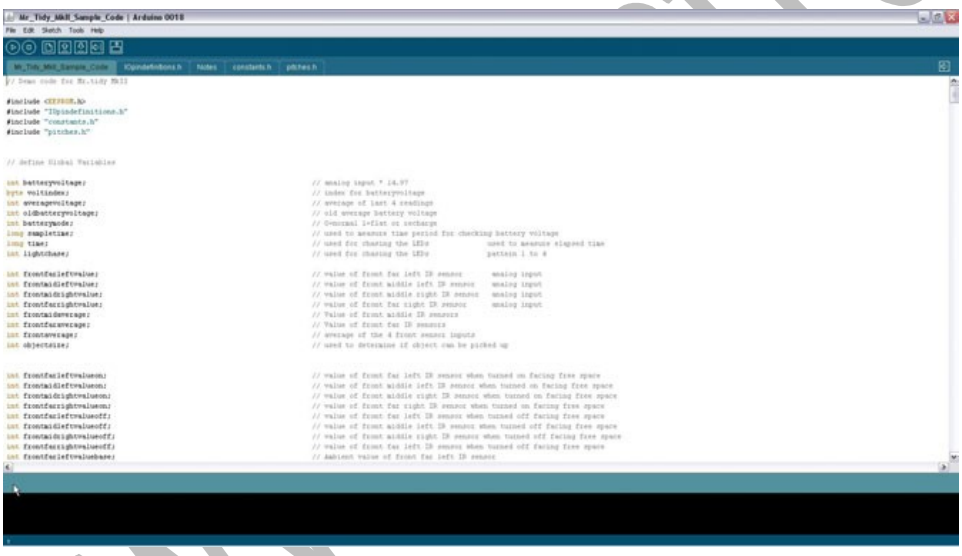


ШАГ 9: Внимательно проверьте все соединения. При включении робота должна раздаться мелодия.

ШАГ 10: Что бы увеличить надежность датчиков могут быть установлены дополнительные кольца, однако они могут привести к небольшому снижению чувствительности.

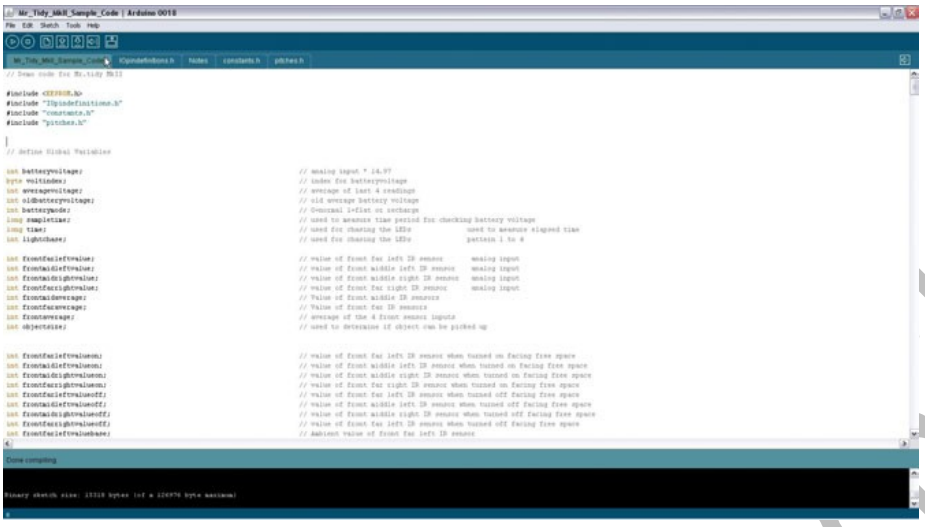
Установка программного обеспечения

Данный робот может быть запрограммирован с помощью бесплатной интегрированной среды разработки, скачиваемой из интернета. В зависимости от вашего опыта в программировании, существует несколько вариантов. Для людей, обладающих начальными навыками, подойдет AVR Studio.
Что бы максимально упростить данного робота, он полностью совместим с Arduino. Для этого на плате предусмотрен USB разъем. Данное руководство предполагает, что вы будете использовать Arduino IDE версии 18 и более поздние. Хотя робот поставляется уже с загрузчиком Arduino и прошивкой, вам понадобится Arduino IDE, что бы использовать диагностическое программное обеспечение и для написания собственного кода.
В настоящее время Arduino IDE доступно Windows, Mac OS X и Linux 32bit. Среду можно скачать по ссылке:
http://arduino.cc/en/Main/Software
После того, как вы установили Arduino IDE на свой компьютер, вы можете открыть программу “Mr_Tidy_Sample_Code.pde”, которая поставляется вместе с роботом, либо вы можете скачать ее по ссылке:
http://arexx.com.cn/en/DownList.asp

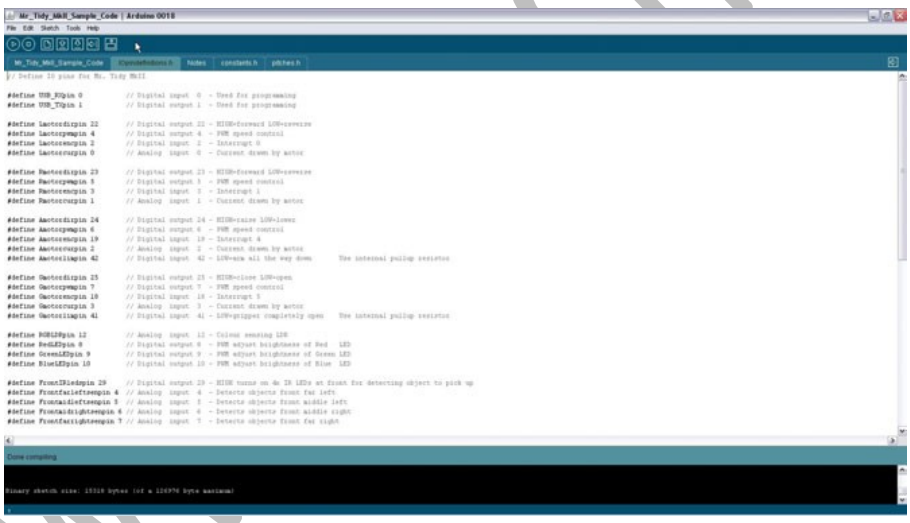


Краткий обзор программного обеспечения

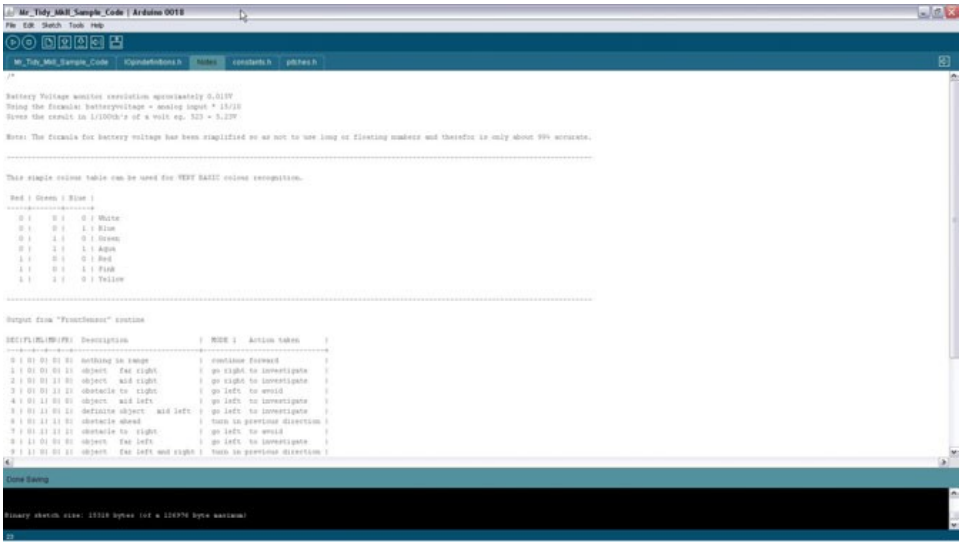
Программа, поставляемая с роботом, реализует алгоритм поиска объектов, и сортировки их по цвету. Это всего лишь простая демонстрационная программа, но она содержит инструкции для работы всех моторов и сенсоров. Данный пример написан с помощью среды разработки Arduino, так как она наиболее проста для начинающих. Вы увидите несколько вкладок в верхней части окна, которые помогают облегчить процесс написания кода. На первой вкладке отображается исходный код главной программы. Вы можете редактировать ее в этом окне.



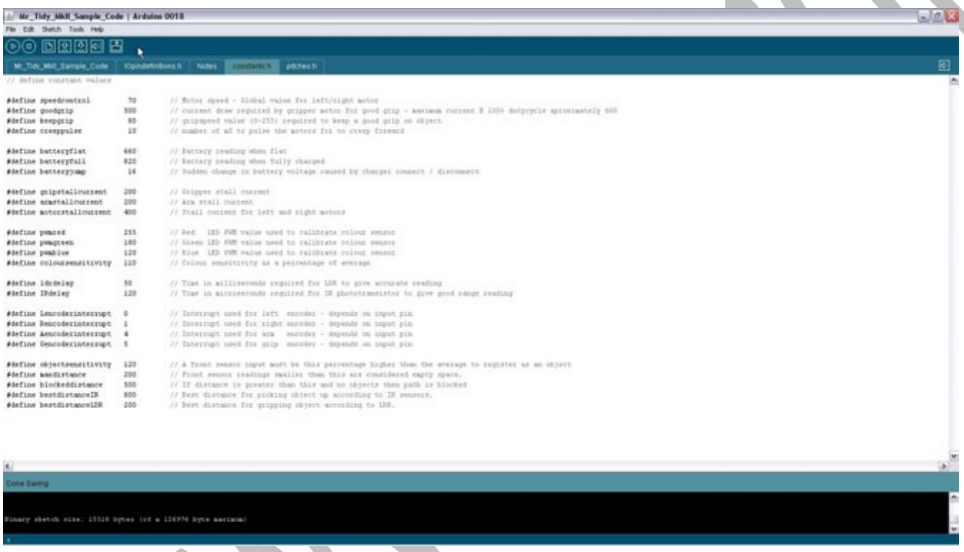
На второй вкладке определены порты общего назначения робота. Atmega1280 имеет 70 портов, из которых более чем 40 используются для управления сенсорами и моторами. Если вы подключите дополнительные устройства, то вы можете определить новые порты тут. Содержимое этой вкладки представляет схему робота.



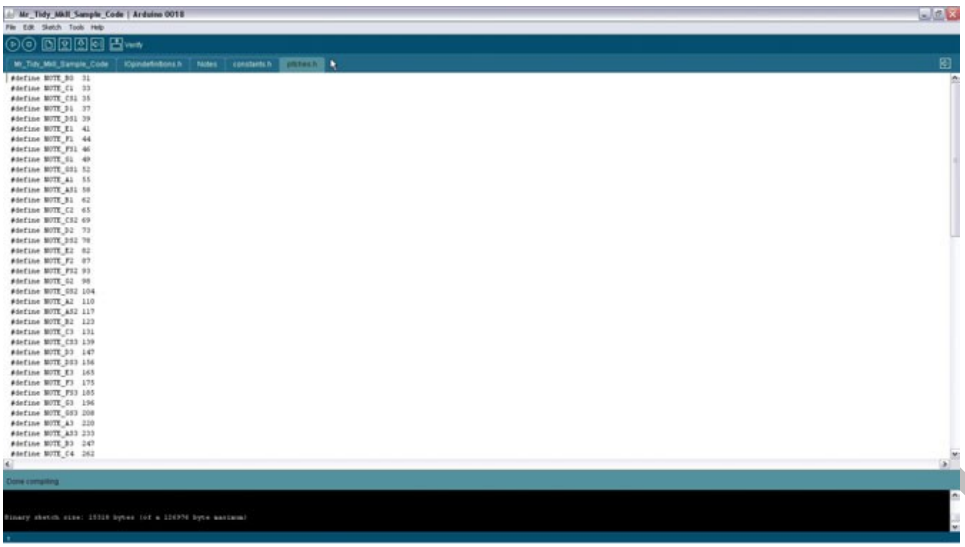
Третья вкладка содержит информацию о том, как некоторые из сенсоров используются в программе. Если вы пишете свой собственный код, вы можете добавить больше информации. Это просто странница комментариев.



Четвертая вкладка содержит список констант. Обычно данные значения не меняют. Они расположены здесь, что бы облегчить отладку программы, например градуировку RGB сенсора.
Гораздо проще поменять значение в одном месте, чем по всей программе, там, где оно встречается.



Последняя вкладка содержит таблицу преобразования музыкальных нот, за что отдельная благодарность Бретту Хэгмену. Эта таблица может быть использована для создания простых мелодий.



Структура программы:
Программа начинается с включения файлов, которые отображены во вкладках, а так же всех необходимых библиотек. Затем определяются глобальные переменные, доступ к которым возможен из любой точки программы.
Функция **setup()** необходима для настроек робота во время запуска. Она вызывается при каждом включении робота по одному разу. Конфигурируются прерывания и порты, которые будут в режиме выходов. По умолчанию, все порты находятся в режиме входов, что снижает вероятность случайного короткого замыкания. После того, как отыграет мелодия, сигнализирующая о готовности робота, манипулятор примет свое исходное положение и робот будет готов к дальнейшим манипуляциям.
Функция **loop()** является ядром программы. Ее содержимое будет постоянно повторяться по кругу, пока робот не будет выключен или перезагружен. Алгоритм этой функции может быть разбит на 7 шагов:
1. Проверка таймера. Если необходимо – изменить параметры поиска.
2. Отслеживать заряд аккумулятора и подзаряжать его, если необходимо (при подключенном источнике питания).
3. Завершение работы робота, если источник питания подключен (после перезарядки нажмите reset).
4. Проверка боковых датчиков, для предотвращения столкновений.
5. Проверка передних датчиков.
6. Выполнить одно из действий, в зависимости от показаний передних датчиков.
7. Определить, если робот уронил объект и сбросить значения некоторых переменных, если необходимо.
Выполняя постоянно эти семь простых шагов, робот будет обнаруживать, и подбирать объекты, пытаясь отсортировать их согласно цвету. Так как данная программа является простой демонстрацией, то функция расстановки не включена и робот будет просто двигаться в случайном направлении, подбирать и ставить объекты. Все функции, используемые в программе – доступны.
As this is a simple demonstration program there is no mapping involved. Информация о языке программирования может быть найдена тут:
http://arduino.cc/en/Reference/HomePage (на английском языке)
http://arduino.ru/Reference

Принципы работы

Прежде, чем вы сможете написать вашу собственную программу для Мистера Тайди, необходимо понять, как устроены сенсоры и как нужно управлять моторами.

Передние сенсоры:
Каждый передний сенсор состоит из одного инфракрасного светодиода и двух инфракрасных фототранзисторов. Оба фототранзистора соединены параллельно, что бы увеличить чувствительность. Данные сенсоры являются аналоговыми. Отраженная инфракрасная волна будет влиять на их показания. Приложению необходимо отградуировать сенсор. Для этого делается два контрольных измерения.
Первое значение измеряется при включенных инфракрасных светодиодах и зависит от внешней засветки и от того, как излучение светодиодов отражается от близлежащих объектов. Чем больше света, тем больше значение.
Второе значение измеряется при выключенных инфракрасных светодиодах. Данная величина зависит только от освещенности помещения. Путем вычитания величину освещенности из итогового значения, вы получаете значение, эквивалентное излучению, отраженному от близлежащих объектов.
В программе вы увидите, что при перемене состояний инфракрасных светодиодов используется небольшая временная задержка (IRdelay). Она рекомендована для получения лучшей точности от фототранзисторов. Величина задержки установлена во вкладке констант и при необходимости может быть изменена.
Следует избегать яркого солнечного света, так как он приведет к снижению чувствительности сенсоров.

Боковые и задние сенсоры:
Боковые и задние сенсоры схожи с передними сенсорами, но они имеют только один фототранзистор. Поэтому их чувствительность примерно в 2 раза меньше. Данные сенсоры тоже аналоговые, но подключены к цифровым портам, так как нам необходимо знать находится ли объект поблизости и не более. Внешняя засветка будет влиять на датчики и, возможно, приведет к ложным показаниям, если освещение будет слишком ярким.
Боковые и задние сенсоры так же имеют светодиоды видимого участка спектра, подключенные параллельно инфракрасным светодиодам. Эти светодиоды управляются независимо от сенсоров, и служат для отображения модели поведения или состояния робота.

Инфракрасный приемник:
Инфракрасный приемник – полностью цифровой сенсор. В отличие от других сенсоров его задача – обнаруживать инфракрасное излучение, модулированное с частотой 38кГц. Он игнорирует внешнее излучение. В нормальном состоянии на выходе этого сенсора всегда высокий уровень напряжения (5В). Когда приемник обнаруживает инфракрасное излучение, модулированное с частотой 38кГц, напряжение на его выходе падает до нуля. Данный сенсор идеален для обнаружения сигналов от пульта дистанционного управления, или инфракрасного маячка. Если необходимо наладить связь между двумя роботами, этот сенсор может принимать данные от другого робота, который модулирует сигнал, с частотой 38кГц.

Распознавание цвета:
Данный робот использует трехцветный светодиод и фоторезистор, что бы распознавать цвета. Идея состоит в том, что бы освещать объект разными цветами. Измерив насколько много излучаемого света отразилось обратно, мы определяем цвет объекта. В приведенной программе мы измеряем напряжение на фоторезисторе, освещая объект красным, зеленым и синим цветами. Так же необходимо измерить напряжение при выключенном цветном светодиоде, что бы исключить фоновое излучение.
Программа сравнивает полученные результаты со средними значениями для красного, зеленого и синего цветов, что бы получить простой трехбитовый результат. Это не самый точный метод, но для данной задачи вполне подходит. Сенсор очень удобно калибровать, так как цвета на светодиоде генерируются с помощь широтно-импульсной модуляции.

Управление моторами:
Каждый из четырех моторов робота управляется “H” мостом. Данная схема обычно используется для управления DC моторами. “H” мост служит для того, что бы при поступлении на него слаботочного сигнала от микроконтроллера, подать на мотор такой же сигнал, но большей мощности. Что бы измерять поворот моторов, установлены специальные оптические сенсоры.
Каждый мотор имеет два управления и 2 выхода для обратной связи. Они определены во вкладке “IOpindefinitions.h”. Рассмотрим, для примера, левый мотор:

Lmotordirpin регулирует направление левого мотора. Когда на порте логическая единица – колеса крутятся вперед, когда логический ноль – в обратную сторону.

Lmotorpwmpin регулирует скорость левого мотора, с помощью широтно - импульсной модуляции. В среде разработки Arduino используется команда analogWrite() для генерации ШИМ, который варьируется от 0 до 255. Ноль соответствует полной остановке мотора, а 255 – вращению мотора на полную мощность.
На небольшой скорости мотор может остановиться. Простой способ предотвратить остановку мотора на низкой скорости – чередовать необходимую вам скорость с наименьшей скоростью, на которой мотор работает без сбоев. Например, если вы хотите величину скорости 23 и вы знаете что мотор работает, начиная с 60 и выше, то организуйте программу так, что бы она меняла скорость в интервале между 23 и 60.

Lmotorencpin регистрирует цифровой сигнал от оптического датчика оборотов, который распознает черные и белые метки на редукторе мотора. На выходе датчика оборотов будет логическая единица, если полоса черная, и логический ноль, если полоса белая. Что бы считать количество полос, программа использует внешние прерывания.

Lmotorcurpin аналоговый сигнал, который отображает ток, потребляемый мотором. Используя команду analogRead() среда разработки Arduino будет возвращать значения. Например, величина сигнала, равная 400 приблизительно соответствует потребляемому току, величиной 1А. Это может быть полезным не только для определения тока, при котором мотор останавливается, но так, же для профилактики робота. Например, если пыли или грязь попала между шестеренками, робот, вероятно, продолжит работу, но средний потребляемый ток мотором возрастет.

Amotorlimpin и **Gmotorlimpin** являются цифровыми входами, используемые для манипулятора. С помощью подтягивающих резисторов, цифровые входы по умолчанию находятся в состоянии логической единицы. Значение логического нуля сигнализирует о том, что выключатель замкнут

Использование обратной связи мотора:
Следя за датчиками вращения и датчиками тока робота, можно получить много полезных функций. Датчики вращения колес позволяют измерять скорость робота и пройденную дистанцию, что бы создать карту местности, которая поможет роботу лучше ориентироваться в пространстве.
Ток, протекающий через мотор (отвечающий за удержание) манипулятора во время подъема, можно использовать, что бы оценить вес объекта и определить, нужно ли приложить большее давление, что бы удержать груз. Так же с помощью датчика вращения мотора, который отвечает за поднятие груза манипулятором, можно определить движется манипулятор или остановился под весом тяжелого объекта. В таком случае можно увеличить ток, протекающий через мотор, после чего снова проверить состояние мотора. Если началось вращения мотора и скорость подъема удовлетворяет поставленной задаче, то приложенный ток достаточен, либо же следует его увеличить еще.

Устранение неполадок: Если робот работает неправильно, пожалуйста, прочтите данную информацию. Во многих случаях вам понадобится источник питания на 12В (500мА или больше), подключенный к разъему зарядки. Так же, вероятно, вам потребуется установить диагностическое программное обеспечение, которое вы можете скачать с нашего сайта, по ссылке:
http://arexx.com.cn/en/DownList.asp

Примечание: не следует использовать робота при ярком солнечном свете, так как чувствительность сенсоров может снизиться.

Проблема – выключатель замкнут, но робот не работает.
Решение – проверьте, подключен ли аккумулятор. Подключите источник питания на 12В постоянного тока (не меньше, чем 500мА), в разъем зарядки аккумуляторов и нажмите кнопку reset. Если программа загружена в память робота, то перед началом движения должна проиграться мелодия. Если вы используете собственную программу, то вам следует загрузить диагностическую программу и протестировать плату с подключенным к разъему зарядки источником питания.

Проблема – датчики вращения не работают.
Решение – Проверьте, что кабели датчиков вращения подключены правильно и все винты возле датчиков завернуты плотно. Так же проверьте, что бы диск был наиболее близко расположен к датчику на плате. Загрузите диагностическую программу и подключите источник питания в разъем зарядки. Проверьте, работает ли датчик вращения с диагностической программой и, если необходимо, отрегулируйте положение диска.

Проблема – у робота есть проблемы с распознаванием объектов и захватом.

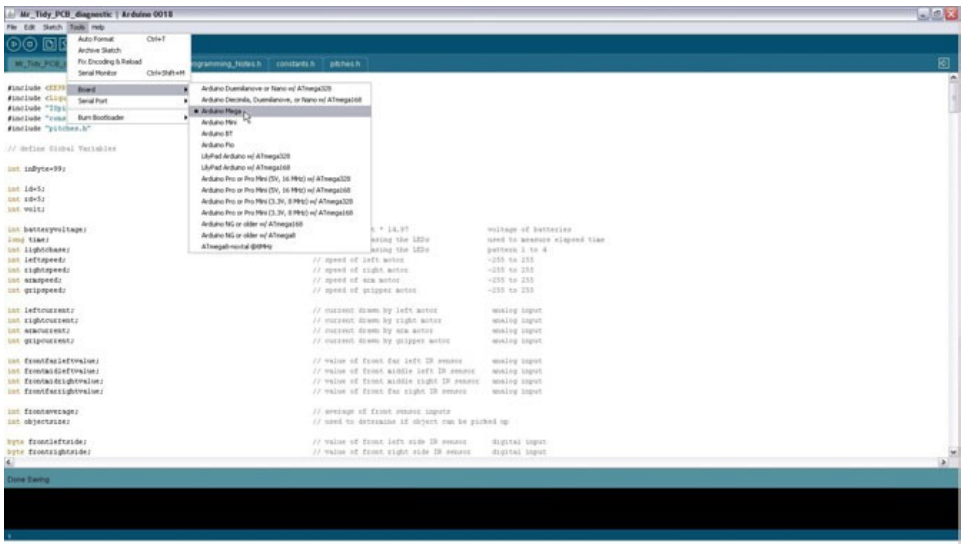
Решение – загрузите диагностическую программу и отрегулируйте положение каждого фототранзистора, так что бы каждый сенсор возвращал примерно одинаковое значение (примерно 50), когда поблизости нет никаких объектов.

Проблема – один из датчиков, предотвращающих столкновение, не работает.
Решение – датчик требует повторной настройки. Используйте программное обеспечение, либо диагностическую программу. Ближайший к сенсору синий светодиод загорится, когда объект будет обнаружен. Инфракрасный светодиод и фототранзистор должны быть параллельны друг другу, но не соприкасаться. Аккуратно настройте их положение, что бы сенсоры обнаруживали объект, когда вы подносите руку, на расстоянии, примерно, 40 мм от сенсора.

Использование диагностического программного обеспечения

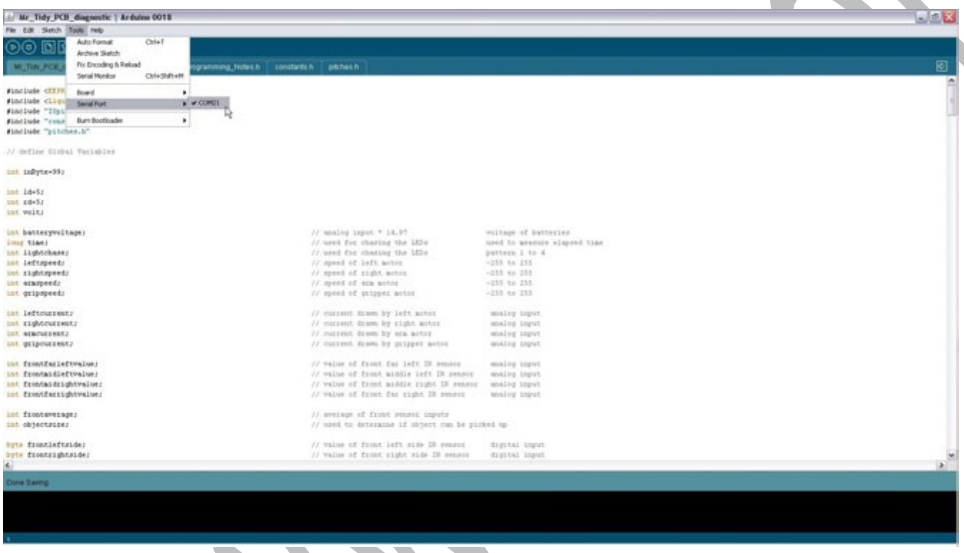
Диагностическое программное обеспечение позволяет вам тестировать все сенсоры робота. Что бы использовать программу вам необходимо установить среду разработки Arduino на ваш компьютер (версию 18 или более позднюю).

Откройте программу Mr_Tidy_PCB_diagnostic. Зайдите в настройки и выберете тип платы “Arduino Mega”.

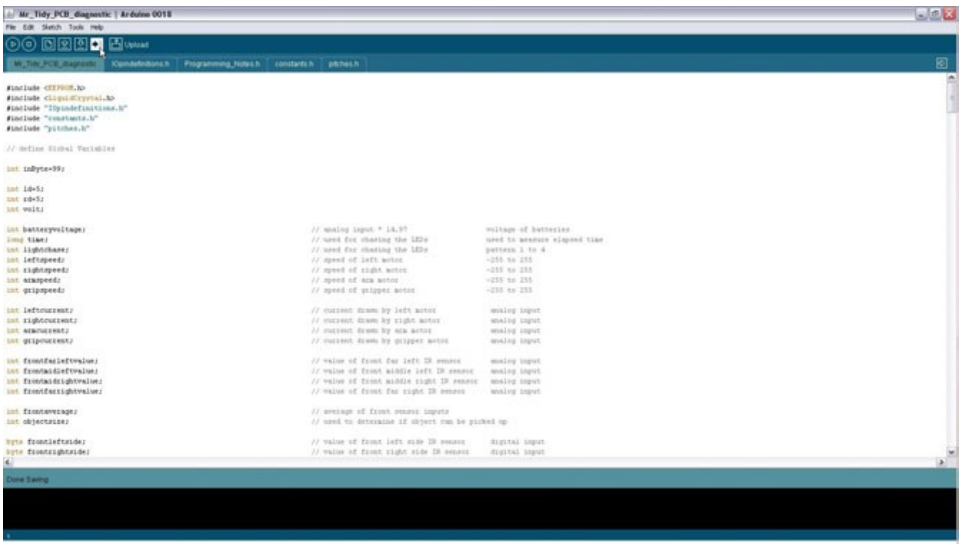


.

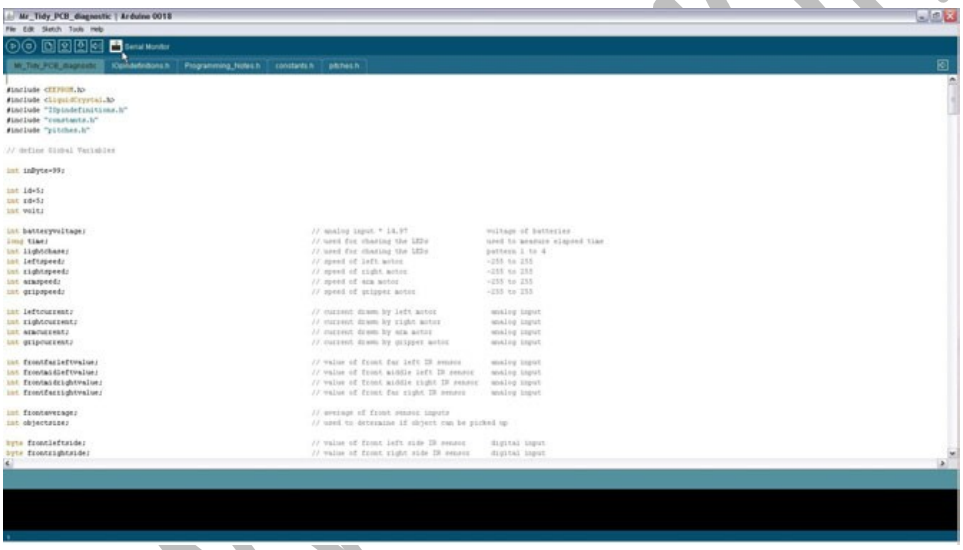
Убедитесь, что робот соединен с компьютером через USB кабель и включен. Только так компьютер сможет установить соединение. Выберете номер последовательного порта.



Загрузите в память робота диагностическую программу. Вы должны увидеть мигание коммуникационных светодиодов и сообщение об успешной загрузке. Если произошла ошибка – проверьте настройки последовательного порта и что робот включен. Если заряд аккумуляторов недостаточен, то придется подключить источник питания в разъем зарядки.

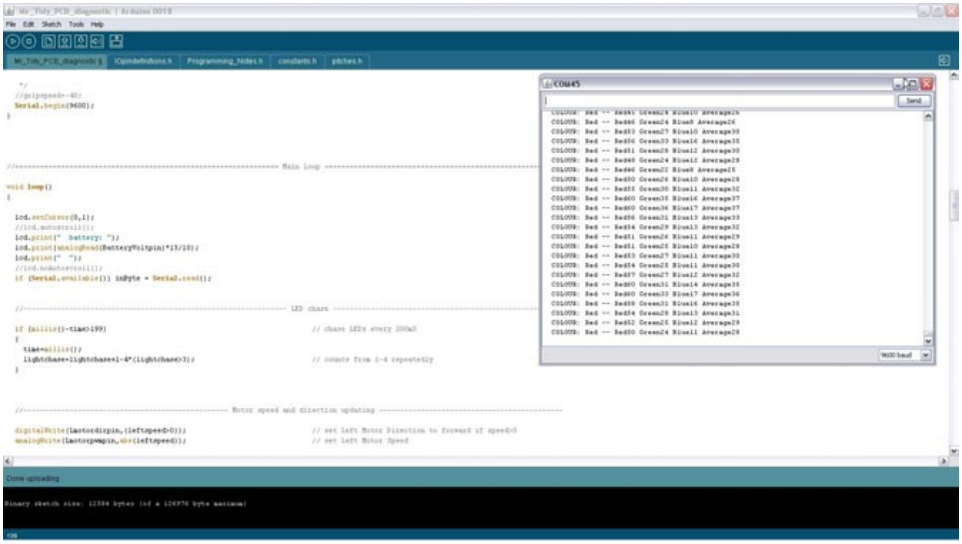


Теперь запустите монитор последовательного порта, который позволяет вам взаимодействовать с роботом. Убедитесь что скорость передачи данных (отображается в правом нижнем углу окна последовательного монитора), установлена на 9600.

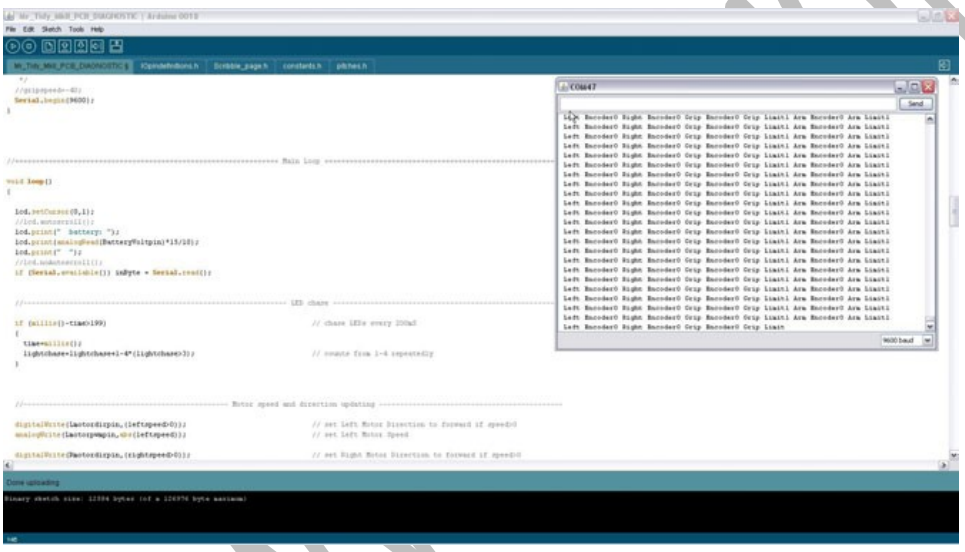


Робот начнет работу с проигрывания небольшой мелодии. Синие светодиоды в углах робота, начнут мерцать. Датчики столкновения можно проверить, поднеся руку на небольшом расстоянии. Когда сенсор обнаруживает вашу руку, соответствующий светодиод перестает мерцать.

Информацию о датчике цвета вы увидите в мониторе последовательного порта. Что бы откалибровать его, поместите кусок белой бумаги напротив робота. Настройте красный, зеленый и синий цвета и перезагрузите программу снова. Когда значения красного, зеленого и синего цвета, передаваемые в монитор последовательного порта, станут одинаковыми, можно считать, что сенсор откалиброван. Сохраните эти значения, так как они могут пригодиться.

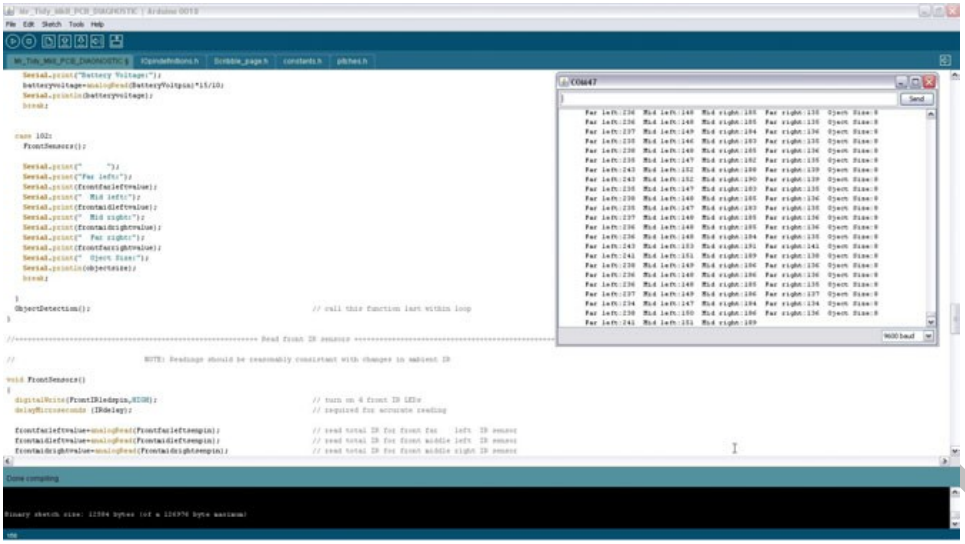


Кликните мышкой в левом верхнем углу монитора последовательного порта, и введите “e”, затем нажмите [enter]. Это действие пошлет команду роботу отобразить на мониторе информацию о датчиках вращения.

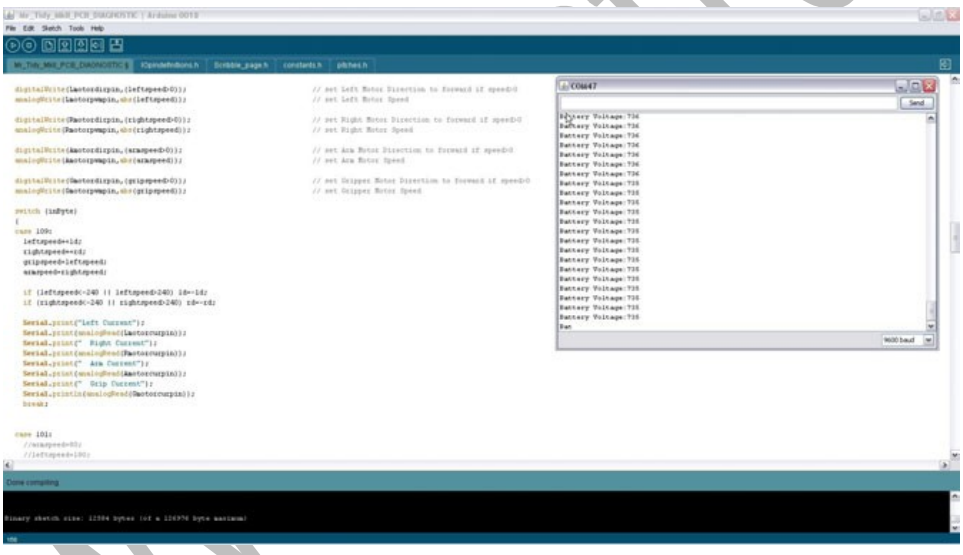


Вращая колеса и манипулятор, вы сможете проверить работоспособность всех датчиков вращения.

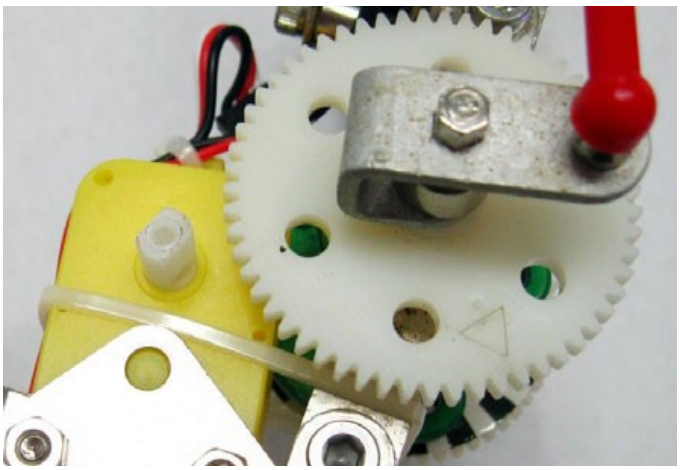
Что бы откалибровать передние датчики, введите “f” и затем [Enter]. После чего на экран будет выведена информация о 4 датчиках. Поместите робота на пустое место, что бы в радиусе 1 метра не было никаких преград. Все датчики должны иметь одинаковое значение. Поднеся руку к сенсору, вы должны увидеть, что значение на экране поменялось.



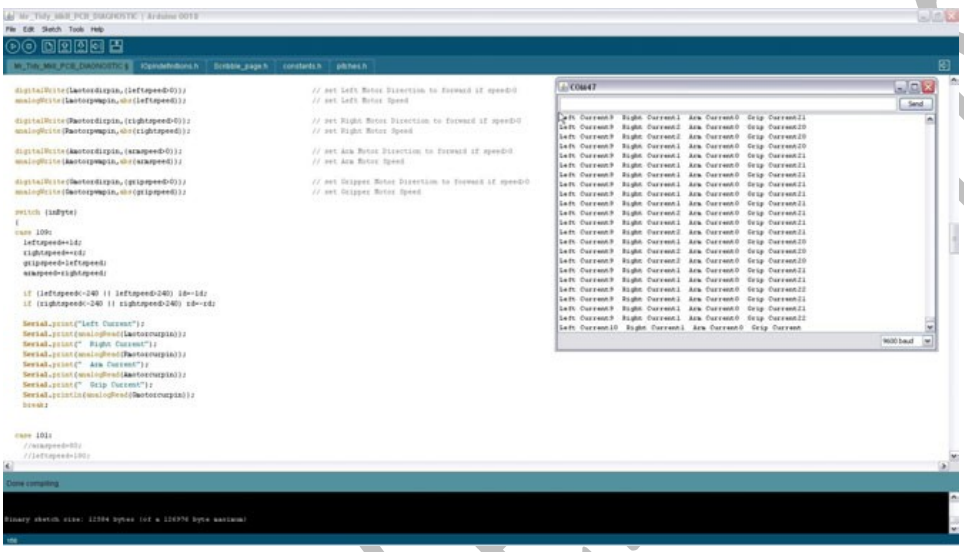
Дополнительные кольца, одеваемые на датчики, могут снизить чувствительность, но они помогают сохранить стабильность робота, если тот встречает препятствие.
Заряд аккумуляторов может быть выведен на экран нажатием “b” и затем [Enter]. Считываемое значение примерно равно 100х.
Реальное значение напряжение в данном случае должно быть около 7.35В. Значение напряжения может немного колебаться, так как включены сенсоры и моторы.



Заключительный тест для “H” мостов, которые управляют моторами. Когда этот тест запущен все моторы вращаются вперед и назад без учета ограничений. Рекомендуется снять редукторы с моторов манипулятора. Если вам необходимо проверить только один мотор, то отключите все остальные от платы.



Теперь, когда робот подготовлен, нажмите “m” и [Enter], что бы начать тест.



Характеристики

Микроконтроллер:
Atmega1280-16AU
Тактовая частота:
16МГц
Память:
128K FLASH (2K использовано под загрузчик), 4K EEPROM, 8K SRAM
Поддерживаемые интерфейсы:
USB, I2C, SPI, TTL serial, IR
Частота работы инфракрасного сенсора:
38кГц
Драйвер моторов:
4x FET “H” моста рассчитанные на максимальный ток 4A
2x датчика вращения– разрешение: 24 отсчета на оборот
1x датчик поднятия манипулятора – разрешение: 27 отсчетов на 90 градусов
1x датчик сжатия – разрешение: 12 отсчетов от полного сжатия до отпускания
Питание:
6x NiMh “AA” или “UM3” NiMh или NiCd аккумуляторы
Разъем зарядки (2.5мм) - требуется 12В DC (500мА)