

2 СИСТЕМНОЕ ПРОЕКТИРОВАНИЕ

2.1 Определение структуры комплекса

Целью проекта является разработка подвижной платформы и управление ей с помощью программного средства.

В разрабатываемой системе можно выделить следующие блоки:

- центральный контроллер;
- блок обмена данными;
- блок беспроводной связи;
- блок управления;
- блок отображения данных;
- блок управления моторами;
- блок питания;
- блок определения местоположения.

Структурная схема, иллюстрирующая перечисленные блоки и связи между ними приведена на чертеже ГУИР.400201.050 С1.

Из данных требований можно сделать вывод о необходимых элементах структурной схемы.

Рассмотрим каждый блок отдельно.

2.2 Центральный контроллер

Центральный контроллер представляет собой контроллер на плате Arduino Uno. Основной задачей данного блока является управление всеми нижеперечисленными блоками: обработка приходящих от них сигналов, принятие решения об изменении состояния системы и само изменение состояний проектируемого объекта.

Управляющий блок взаимодействует напрямую практически со всеми остальными блоками, т.к. является основным вычислительным модулем. На корпусе управляющего блока будет находиться также модуль беспроводного соединения, благодаря которому устройство может быть управляемо по локальной сети с компьютера.

Управляющий блок будет оснащен картой памяти, которая будет кэшировать приходящую информацию, находящейся вне самого устройства.

Также интересной особенностью может являться подключение к устройству напрямую к управляющему блоку для перепрошивки и других настроек. Тем не менее это будет работать и по локальной сети.

2.3 Блок обмена данными

Блок обмена данными необходим для обмена данными между аппаратной частью API микроконтроллера и пользовательской программой.

Является связующим звеном между блоком беспроводной связи и блоками отображения данных и управления. Данный блок предоставляет удобный процесс обмена данными.

2.4 Блок беспроводной связи

Блок беспроводной связи необходим для дистанционного управления платформой с помощью программного средства на компьютере.

Используется модуль ESP8266. Управляющее устройство общается с ESP8266 через UART (Serial-порт) с помощью набора AT-команд. Поэтому работа с модулем тривиальна для любой платы с UART-интерфейсом: использовать можно Arduino, Raspberry Pi, что душе угодно.

Работа над приёмом и передачей данных выглядит, как взаимодействие с сырым TCP-сокетом или с serial-портом компьютера.

Более того, модуль можно перепрошивать. Программировать и загружать прошивки можно через Arduino IDE, точно так же, как при работе с Arduino. Реакция на AT-команды — это просто функция штатной прошивки, устанавливаемой на заводе. А вы можете написать свою собственную, если того требует проект. Поскольку на модуле есть 2 порта ввода-вывода общего назначения, вы можете обойтись вовсе без управляющей платы: просто подключите периферию непосредственно к ним.

Для того, чтобы среда Arduino IDE научилась прошивать ESP8266 достаточно добавить директорию с конфигурацией платформы в папку со своими скетчами.

Для физического соединения при прошивке вам понадобится USB-Serial адаптер или плата Arduino/Iskra, настроенная в режим USB-моста.

ESP8266 может работать как в роли точки доступа так и оконечной станции. При нормальной работе в локальной сети ESP8266 конфигурируется в режим оконечной станции. Для этого устройству необходимо задать SSID Wi-Fi сети и, в закрытых сетях, пароль доступа. Для первоначального конфигурирования этих параметров удобен режим точки доступа. В режиме точки доступа устройство видно при стандартном поиске сетей в планшетах и компьютерах. Остается подключиться к устройству, открыть HTML страничку конфигурирования и задать сетевые параметры. После чего устройство штатно подключится к локальной сети в режиме оконечной станции.

В случае исключительно местного использования возможно всегда оставлять устройство в режиме точки доступа, что снижает необходимые усилия пользователя по его настройке.

После подключения к Wi-Fi сети устройство должно получить IP-параметры локальной сети. Эти параметры можно задать вручную вместе с параметрами Wi-Fi либо активизировать какие-либо сервисы автоматического конфигурирования IP-параметров (например, DHCP).

После настройки IP параметров обращение к серверу устройства в локальной сети обычно осуществляется по его IP адресу, сетевому имени (в случае если имена поддерживаются какой-либо технологией, например NBNS) или сервису (в случае если поддерживается автоматический поиск сервисов, например через протокол SSDP).

2.5 Блок управления

Для настройки устройства под конкретного пользователя необходим удобный интерфейс. Блок управления будет представлять собой набор инструментов для программного средства, с которым взаимодействие происходит по локальной сети. Пользователь всегда сможет зайти в программу, подключиться и начать взаимодействовать с платформой, посылая ей команды. Команды управления будут представлять собой набор команд для получения интересующей пользователя информации, а также возможность управления устройством. Например, задавать направления движения платформы, получать координаты местоположения платформы и прочее.

Каждая команда сформирована таким образом, чтобы данный блок сразу преобразовал ее в предпочтительный для себя вид и в соответствии с тем, что пришло, выдает конкретные команды. Тем самым является обработчиком приходящих команд от блока обмена данными.

2.6 Блок отображения данных

Блок отображения данных необходим для отображения данных, которые приходят от блока обмена данными, если блок управления распознает, что это результат пришедшей команды, предназначенной для отображения на экран. Данный блок будет зависеть от блока управления, так как блок управления определяет тип команды, пришедшей от блока обмена данными, и проверяет, следует ли ее передавать на блок отображения данными.

2.7 Блок управления моторами

Блок управления моторами предназначен для регулирования скорости моторов, следит за корректной работой отдельных моторов и т.д.

Блок управления моторами выполняет крайне важную роль в проектах ардуино, использующих двигатели постоянного тока или шаговые двигатели. В данном проекте использовался популярный драйвер двигателей на базе микросхемы L298N.

Как известно, плата Arduino имеет существенные ограничения по силе тока присоединенной к ней нагрузки. Для платы это 800 mA, а для каждого отдельного вывода — и того меньше, 40mA. Мы не можем подключить

напрямую к Arduino Uno, Mega или Nano даже самый маленький двигатель постоянного тока. Любой из этих двигателей в момент запуска или остановки создаст пиковые броски тока, превышающие этот предел.

Модуль используется для управления шаговыми двигателями с напряжением от 5 до 35 В. При помощи одной платы L298N можно управлять сразу двумя двигателями. Наибольшая нагрузка, которую обеспечивает микросхема, достигает 2 А на каждый двигатель. Если подключить двигатели параллельно, это значение можно увеличить до 4 А.

2.8 Блок питания

Блок питания предназначен для обеспечения корректного и стабильного питания для функционирования платформы.

Большинство плат требует наличие питания в диапазоне от 4.5 до 9 вольт через разъем внешнего питания и 4.5-5 вольт через USB. Однако в инструкции написано 7-12 вольт, то есть будем считать, что оптимальным вариант это 9 вольт.

Питание от 5 вольт. Этот вариант питания от компьютера. Реализовать такое питание можно также от зарядного устройства телефона или купив преобразователь. Если же уже все сделано и плата прошита, то напряжение 5 вольт будет недостаточно. В этом случае при значительных нагрузках на выходы, возможны провалы в работе. Относительно данного проекта 5 вольт недостаточно.

Питание от 9 вольт. Arduino можно запитать от батарейки "Крона" или блока пальчиковых батареек. На холостом ходу или с минимальной нагрузкой она проработает не один месяц. А вот уже с небольшим увеличением нагрузки время автономной работы быстро сойдет на нет. Если увеличить нагрузку на батарейку в виде датчиков требующих больше мощности и светодиодов индикации, то батарейки может хватить совсем ненадолго.

2.9 Блок определения местоположения

Блок определения местоположения отвечает за определение местоположения относительно заданной системы координат.

Пока основной проблемой всех ныне существующих мобильных аппаратов, перемещающихся самостоятельно, без управления со стороны человека, остается навигация. Для успешной навигации в пространстве бортовая система робота должна уметь строить маршрут, управлять параметрами движения (задавать угол поворота колес и скорость их вращения), правильно интерпретировать сведения об окружающем мире, получаемые от датчиков, и постоянно отслеживать собственные координаты.

Полноценный робот должен определять собственные координаты и выбирать направление движения только на основании показателей бортовых

датчиков, поэтому системы искусственного интеллекта, создаваемые для автономных машин, ориентированы на поддержку непрерывного цикла "опрос датчиков – принятие оперативного решения об изменении маршрута". Таких циклов может быть несколько – один ответственен за следование по основному маршруту, другой – за обход препятствий и т. д. Кроме того, на аппаратном уровне каждый цикл может поддерживаться датчиками разных типов и разных принципов действия, формирующих потоки данных разного объема и интенсивности.

В результате робот начинает теряться в сложной обстановке и на длинных маршрутах, когда надо не просто обходить мелкие препятствия и уклоняться от опасностей на относительно прямом пути, а планировать долгосрочные действия на стратегическом уровне и выполнять ряд вспомогательных задач, которые весьма трудоемки сами по себе. Поэтому современные системы навигации объединяют механизмы как низкоуровневого управления, так и высокоуровневого планирования.

Проблемы, непосредственно связанные с движением на текущем коротком отрезке маршрута, решаются путем простого реагирования на особенности внешней среды, а глобальная система следит за соблюдением общего плана, модифицируя его в случае необходимости, и синхронизирует работу всех подчиненных структур управления.