

СИСТЕМА УПРАВЛЕНИЯ МОБИЛЬНЫМ РОБОТОМ

Рассматривается реализация системы управления мобильным роботом с использованием полнодуплексного протокола связи WebSocket для обмена данными с ботовым компьютером робота в режиме реального времени.

ВВЕДЕНИЕ

Для решения задачи управления мобильным роботом предлагается использовать полнодуплексный протокол связи WebSocket. Данный подход позволяет исключить использование дополнительных дорогостоящих приемопередающих устройств для обмена данными с бортовым компьютером мобильного робота при условии, что со стороны оператора и мобильного робота возможен выход в сеть Интернет.

I. ЭЛЕМЕНТНАЯ БАЗА

В качестве бортового компьютера предлагается использовать активно набирающий популярность одноплатный микрокомпьютер Raspberry Pi 3 по ряду причин: высокая производительность микропроцессора, возможность использования операционной системы Linux, наличие встроенных модулей Wi-Fi, Bluetooth LE (Low Energy – пониженное энергопотребление), наличие Ethernet порта, наличие USB разъемов, наличие разъема для установки бортовой камеры, наличие цифровых портов ввода-вывода с возможностью использования протокола соединения I2C (inter-integrated circuits – связанные интегральные схемы внутри системы) [1]. Недостатком данного выбора является отсутствие внутрисистемного ЦАП (цифроаналогового преобразователя). Для решения этой проблемы предлагается использовать микроконтроллер Arduino Mega2560 семейства ARM, имеющий 8-разрядный ЦАП [2]. Для увеличения уровня управляющего ШИМ (широтно-импульсно модулированного) сигнала используется полномостовой драйвер L298N [3].

II. СТРУКТУРНАЯ СХЕМА ЭЛЕМЕНТОВ МОБИЛЬНОГО РОБОТА

Бортовой компьютер и микроконтроллер подключаются с использованием I2C шины. Цифровые выводы 6, 7, 8 и 9 подключаются к входам драйвера IN 1, IN 2, IN 3, IN 4 соответственно. Выходы OUT 1 и OUT 2 драйвера подключаются к положительной и отрицательной клеммам первого двигателя, а выходы OUT 3 и OUT 4 драйвера подключаются наоборот - к отрицательной и положительной клеммам второго двигателя. Это обуславливается тем, что оба двигателя вращаются по умолчанию в одном направлении. Если развернуть двигатели на 180 градусов, что требуется для установки в корпусе робота, при стан-

дартном подключении робот начнет вращаться вокруг вертикальной оси, что неприемлемо.

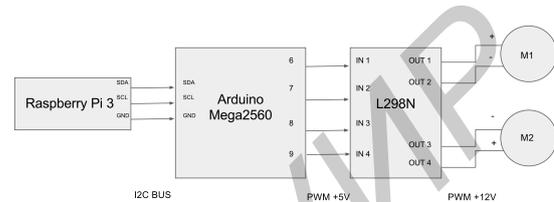


Рис. 1 – Структурная схема мобильного робота

III. ИСПОЛЬЗОВАНИЕ ДЖОЙСТИКА

В качестве управляющего элемента предлагается джойстик. С программной реализацией джойстика можно ознакомиться по ссылке [4].

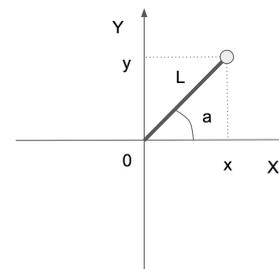


Рис. 2 – Проекция джойстика на горизонтальную плоскость

На рисунке 2 представлен пример использования данных, полученных с джойстика. Ось ординат отвечает за величину линейной скорости, а ось абсцисс - за величину круговой скорости относительно вертикальной оси робота. При использовании выше указанной библиотеки джойстика, можно получить параметр L - длина проекции джойстика на горизонтальную плоскость, и параметр a - угол между проекцией на горизонтальную плоскость и осью абсцисс. По этим параметрам можно определить действующие значения линейной: $y = L \times \sin(a)$, и круговой: $x = L \times \cos(a)$ скоростей.

IV. КИНЕМАТИЧЕСКАЯ ЧАСТЬ СИСТЕМЫ

На рисунке 3 схематично представлена кинематика системы.

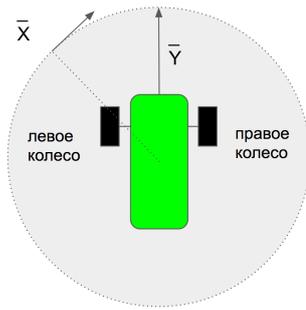


Рис. 3 – Векторы линейной и круговой скоростей

Общепринято считать вращение положительным, если оно направлено против часов стрелки. Однако использовать такое положительное направление вектора круговой скорости, то при перемещении джойстика в сторону по горизонтали частота вращения двигателя с соответствующей стороны начнет увеличиваться. Такое управление не совсем удобно для восприятия человеком. Предлагается реализовать такое управление, которое схоже с управлением автомобилем: при повороте руля вправо автомобиль поворачивает вправо, при повороте налево - налево. Этим и обуславливается выбор положительного направления вектора круговой скорости по часовой стрелке. По данной схеме можно определить скалярные значения скоростей правого: $W_R = x - y$, и левого: $W_L = x + y$ двигателя.

V. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ

В нашей системе можно выделить три основные программные части: интернет приложение, программа бортового компьютера и программа микроконтроллера. Т.к. двигатель постоянного тока с прикрепленным на его вал колесом является довольно инерционным элементом в системе, то нет необходимости в непрерывной передаче данных о положении джойстика. Было решено передавать данные о положении джойстика с интервалом в 100мс. Вторая причина, по которой не советуются непрерывно передавать данные о положении джойстика - микроконтроллер. Дело в том, что бортовой компьютер сможет обработать такой поток данных, а микроконтроллер - нет. В процессе разработки системы мы столкнулись с данной проблемой. На рисунке 4

Теханов Роман Викторович, Белорусского государственного университета информатики и радиоэлектроники, longbeardmen@gmail.com.

Научный руководитель: Ляхор Тимофей Васильевич, магистр технических наук, ассистент, linoge@gmail.com.

Научный руководитель: Чумаков Олег Анатольевич, кандидат технических наук, доцент, olegchumakov@bsuir.by.

представлена кода, отвечающая за вышеописанную логику.

```
manager = nipplejs.create(options),
params = {
  distance: 0,
  angle: 0
},
socket = new WebSocket("ws://192.168.43.198:8000");

manager.on('move', function(event, nipple){
  var angle = nipple.angle.radian,
      distance = nipple.distance;
  params = { angle: angle, distance: distance }
});

setInterval(function(){
  console.log(params)
  socket.send(JSON.stringify(params));
}, 100);
```

Рис. 4 – Логика отправки данных

На рисунке 5 представлена часть кода, отвечающая за логику микроконтроллера.

```
void receiveEvent(int howMany){
  while(1 <= Wire.available()){
    pin = Wire.read();
    pwm = Wire.read();
    analogWrite(pin, pwm);
  }
  Wire.flush();
}
```

Рис. 5 – Логика микроконтроллера

По ранее обработанным бортовым компьютером данным, на микроконтроллер посылаются данные с номером цифрового вывода и значением ШИМ сигнала, которое необходимо получить на этом выводе. Полный программный код системы управления мобильным роботом находится в открытом доступе. С ним можно ознакомиться по ссылке [5]

1. <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>
2. <https://www.arduino.cc/en/Main/arduinoBoardMega2560>
3. <https://www.sparkfun.com/datasheets/Robotics/L298N-Bridge>
4. <https://github.com/yoannmoinet/nipplejs>
5. <https://github.com/longbeardmen/dakillabot>