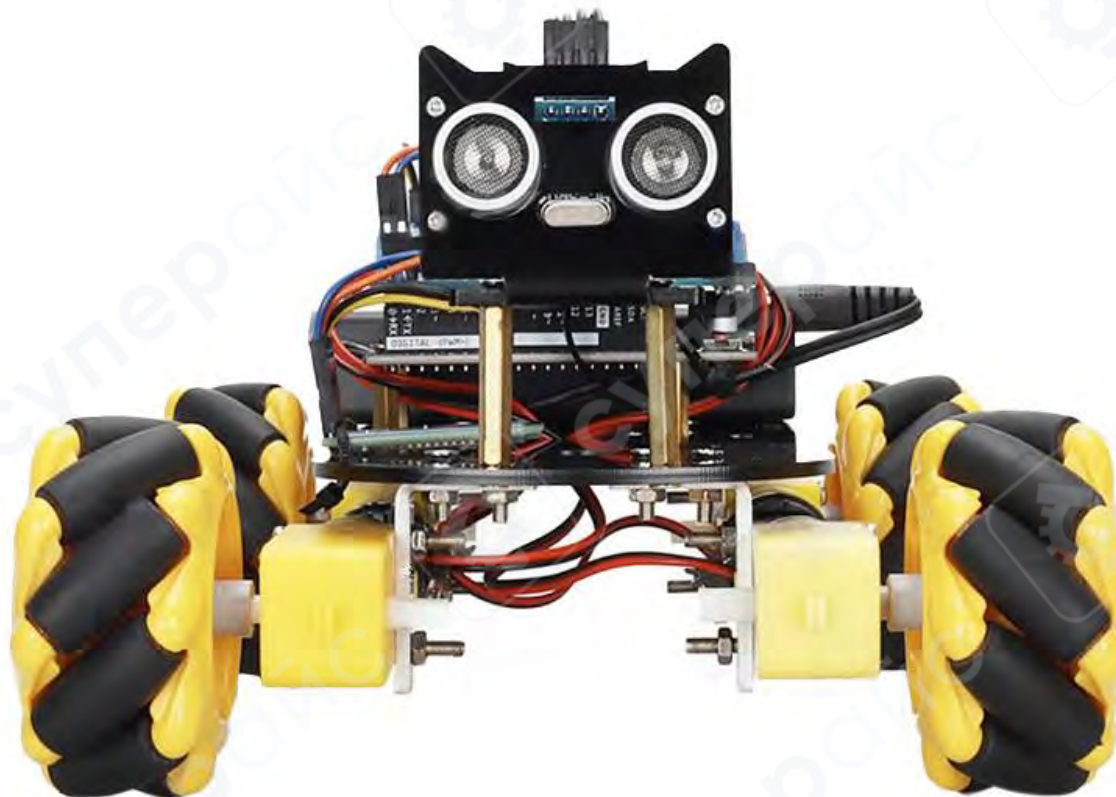


Робот-автомобиль Mini Mecanum Wheel



Инструкция по эксплуатации

Руководство включает три папки, которые пригодятся в процессе знакомства с основами электроники и программирования.

 1_Get_start-准备	2023/4/29 15:04	文件夹
 2_Arduino_Code-代码文件	2023/4/10 19:19	文件夹
 3_Bluetooth_APP	2023/4/12 14:24	文件夹

1_Get_start: содержит инструкцию по сборке робота и файлы среды программирования. Тщательно изучите инструкцию, чтобы собрать робота быстро и правильно.

2_Arduino_Code: содержит скрипты Arduino, которые нужно загрузить на плату управления.

3_Bluetooth_APP: содержит дистрибутив приложения Bluetooth. Информацию о подключении устройства к приложению см. в данной инструкции.

Оглавление

Загрузка кода для управления через Bluetooth.....	5
Установка и настройка приложения	7
Для устройств с Android	7
Для устройств с iOS.....	7
Подключение к устройству для управления через приложение TSCINBUNY	7
Колеса Mecanum	10
Направления сборки колес (вид сверху)	10
Привод и принципы движения	11
Схема подключения колес.....	12
Загрузка скриптов.....	13
Анализ скрипта управления движением тележкой	14
Принципы управления движения тележкой	14
Ультразвуковой дальномер	18
Ультразвуковой датчик	18
Принцип ультразвукового измерения	18
Схема подключения	19
Анализ кода	20
Функция обхода препятствий	21
Анализ кода	21
Управление через Bluetooth.....	22
Модуль Bluetooth	22
Схема подключения	22
Анализ кода	23
Спецификация и сборка.....	24
Комплектующие	24
Установка медных стоек	25
Установка крепежных скоб	25
Установка моторов	26
Установка ультразвукового модуля	26
Установка колес	27
Установка платы разработки и платы расширения	28

Установка блока батареек	29
Установка Bluetooth-модуля	29
Схемы подключения	30

Загрузка кода для управления через Bluetooth

Убедитесь, что среда программирования из папки 1_Get_start установлена.

1_Get_start	2023/6/2 9:54	文件夹
2_Arduino_Code	2023/6/2 9:54	文件夹
3_Bluetooth_APP	2023/6/2 9:54	文件夹

Откройте файл с кодом по адресу: 2_Arduino_Code\MINI_Mecanum_Wheel_Car\MINI_Mecanum_Wheel_Car.ino.

MINI_Mecanum_Wheel_Car.ino	2022/11/11 19:14	INO 文件	8 KB
ino-err.log	2022/11/11 19:49	文本文件	1,546 KB

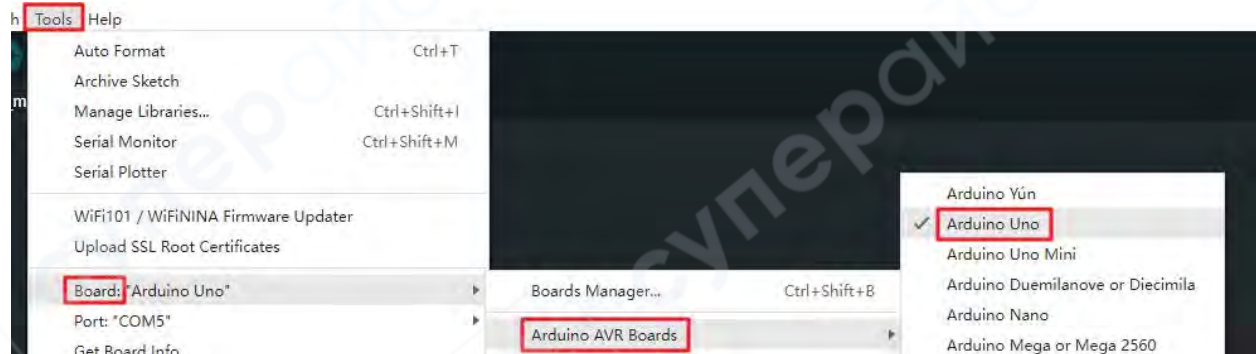
Подключите плату к компьютеру с помощью USB-кабеля.



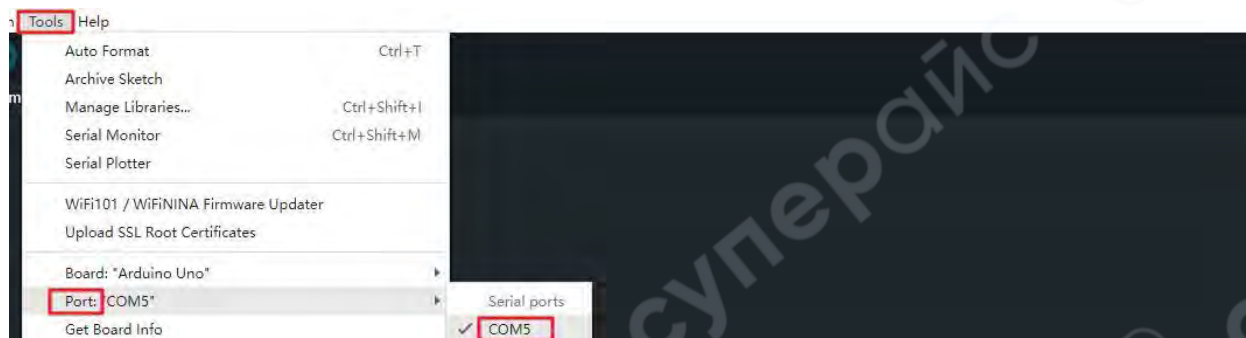
Уберите соединение между RXD/TXD модуля Bluetooth (перед каждой загрузкой необходимо убирать соединение!)



Откройте меню «Инструменты», затем «Плата разработки», выберите Arduino Uno.



Далее выберите последовательный COM-порт.

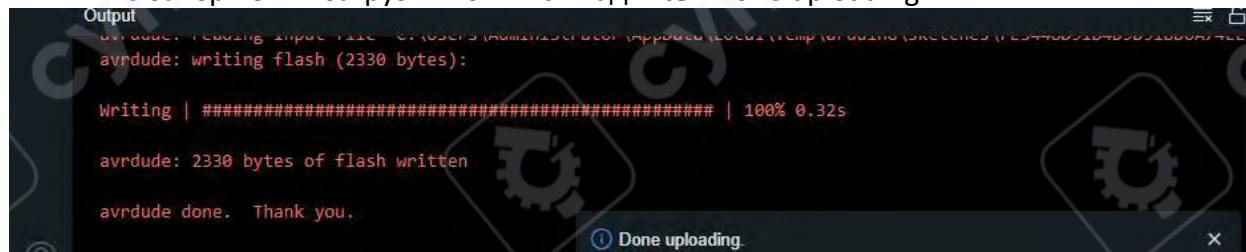


Примечание: в примере указан тип платы управления **UNO** и имя порта **COM5**. Имя портов на разных ПК иногда отличается, и на вашем компьютере нужным портом может быть COM3 и COM4.

Нажмите «Загрузить», чтобы начать загрузку кода на плату управления.



По завершении загрузки появится надпись «Done uploading».



После загрузки верните убранное соединение на Bluetooth-модуле.



Загрузка кода завершена.

Установка и настройка приложения

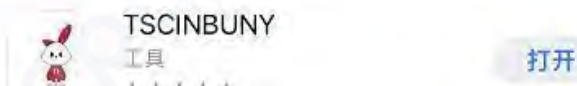
Для устройств с Android

Откройте папку 3_Bluetooth_APP. С помощью инсталлятора Bluetooth.apk установите приложение на смартфон. При возникновении запроса разрешите обновление приложения, следите, чтобы смартфон был подключен к Интернету.



Для устройств с iOS

Откройте App Store на смартфоне, найдите приложение TSCINBUNY.



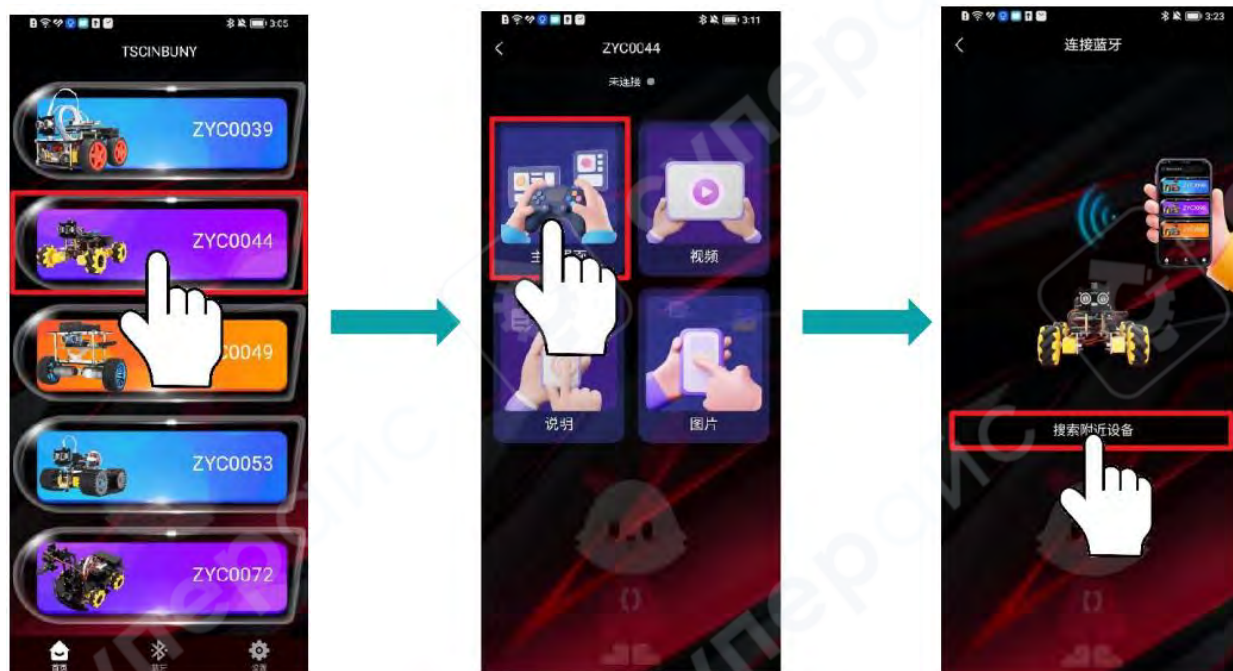
Подключение к устройству для управления через приложение TSCINBUNY

Ниже показано подключение робота к устройству с Android для удаленного управления через приложение.

Включите Bluetooth на устройстве:



Откройте приложение TSCINBUNY для удаленного управления, выберите модель робота-тележки -> Главный интерфейс -> Поиск устройств.



Подключитесь к устройству с именем BT05.





Нажмите кнопку «Подробнее» в верхнем правом углу, чтобы ознакомиться с возможностями управления после успешного подключения тележки к смартфону.

Колеса Mecanum

В этом разделе дано описание электропривода тележки с колесами Mecanum. Колесо Mecanum:



Колеса Mecanum оснащены роликами по всей окружности. Колеса делятся на два типа: «левые» с осевым наклоном роликов влево и «правые» с осевым наклоном вправо. Расположенные под углом ролики частично преобразуют силу на обод колеса в нормальную силу, в результате тележка может двигаться влево и вправо.

Направления сборки колес (вид сверху)

Осевые направления роликов к центру тележки



Привод и принципы движения

Направления вращения колес, при которых тележка движется вправо и влево:



Направления вращения колес, при которых тележка движется вперед и назад:

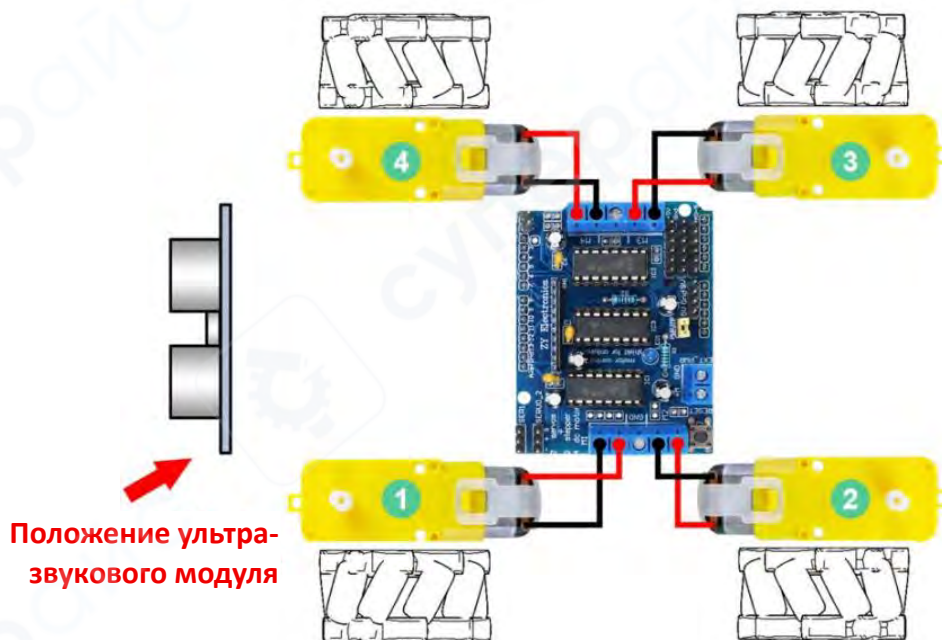


Направления вращения колес, при которых тележка вращается по часовой и против часовой стрелки:



Схема подключения колес

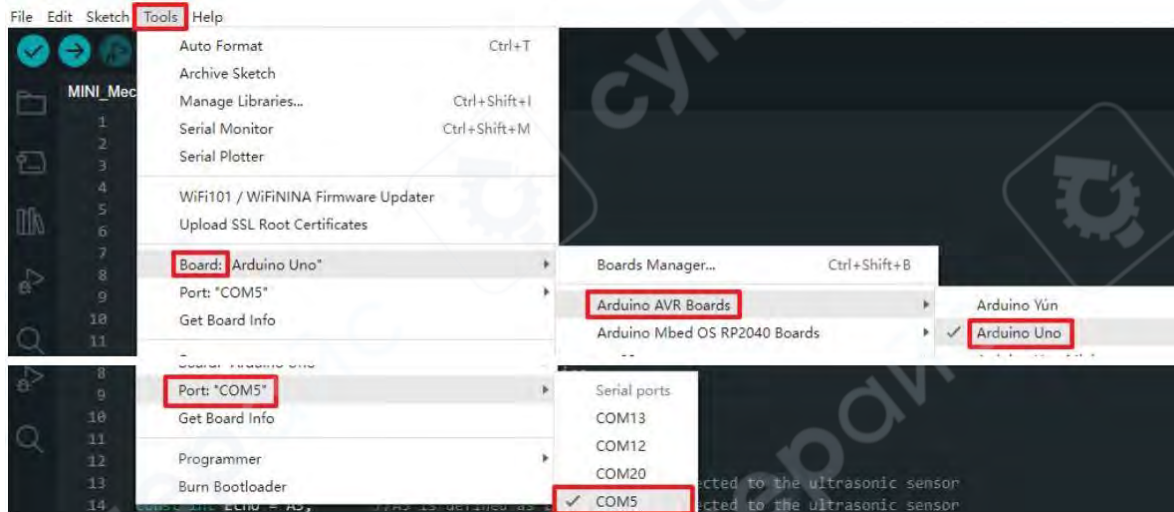
Плата расширения	Мотор
M1	① Левое переднее колесо
M2	② Левое заднее колесо
M3	③ Правое заднее колесо
M4	④ Правое переднее колесо



Загрузка скриптов

Запустите файл с кодом по адресу:
2_Arduino_Code\MINI_Mecanum_Wheel_Car\MINI_Mecanum_Wheel_Car.ino.

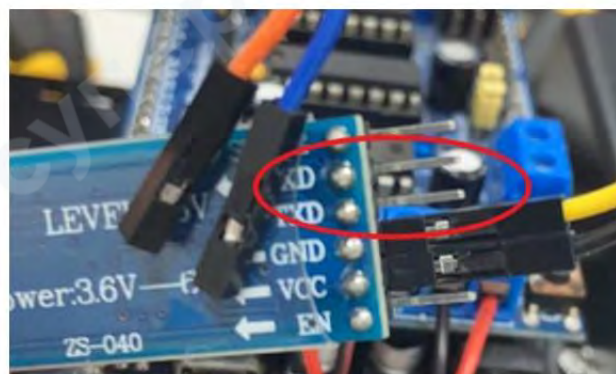
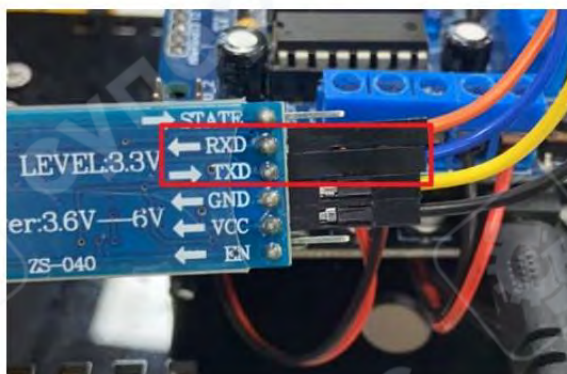
Подключите главную плату управления к ПК, выберите тип платы, затем последовательный COM-порт.



Примечание: в примере указан тип платы управления **UNO** и имя порта **COM5**. Имя портов на разных ПК иногда отличается, и на вашем компьютере нужным портом может быть COM3 и COM4. При подключении к ПК через USB, отображаемый номер COM-порта – это номер COM-порта главной платы.

Нажмите «Загрузить», чтобы начать загрузку скрипта на плату.

Перед каждой загрузкой необходимо убирать соединение TX/RX, после загрузки верните его обратно.



Анализ скрипта управления движением тележкой

Определите пины для управления ШИМ.

```
2 //Configure THE PWM control pin
3 const int PWM2A = 11; //M1 motor
4 const int PWM2B = 3; //M2 motor
5 const int PWM0A = 6; //M3 motor
6 const int PWM0B = 5; //M4 motor
7
8 const int DIR_CLK = 4; // Data input clock line
9 const int DIR_EN = 7; //Equip the L293D enabling pins
10 const int DATA = 8; // USB cable
11 const int DIR_LATCH = 12; // Output memory latch-clock
```

Задайте десятичные значения переменной, соответствующие разным режимам движения тележки.

```
16 const int Forward = 216; //216 save to Forward
17 const int Back = 39; //39 save to Back
18 const int Left = 116; //116 save to Left
19 const int Right = 139; //139 save to right
20 const int Stop = 0; //Parking variable
21 const int L_turn = 198; //turn left
22 const int R_turn = 57; //turn right
```

Задайте четыре переменные мощности ШИМ (скорости движения) со значениями в диапазоне от 0-255. Чем больше значение, тем выше скорость движения. Для лучших показателей движения рекомендуется устанавливать значения больше 120.

```
23 //Set the default speed between 1 and 255
24 int Speed1 = 180;
25 int Speed2 = 180;
26 int Speed3 = 180;
27 int Speed4 = 180;
```

Функция привода Motor зависит от четырех переменных сигнала ШИМ (Speed1-Speed4) и параметра Dir, определяющего направление вращения моторов.

```
65 void Motor(int Dir,int Speed1,int Speed2,int Speed3,int Speed4)
66 {
67     analogWrite(PWM2A,Speed1); //Motor PWM speed regulation
68     analogWrite(PWM2B,Speed2); //Motor PWM speed regulation
69     analogWrite(PWM0A,Speed3); //Motor PWM speed regulation
70     analogWrite(PWM0B,Speed4); //Motor PWM speed regulation
71
72     digitalWrite(DIR_LATCH,LOW); //DIR_LATCH sets the low level and writes the direction of motion in preparation
73     shiftOut(DATA,DIR_CLK,MSBFIRST,Dir); //Write Dir motion direction value
74     digitalWrite(DIR_LATCH,HIGH); //DIR_LATCH sets the high level and outputs the direction of motion
75 }
```

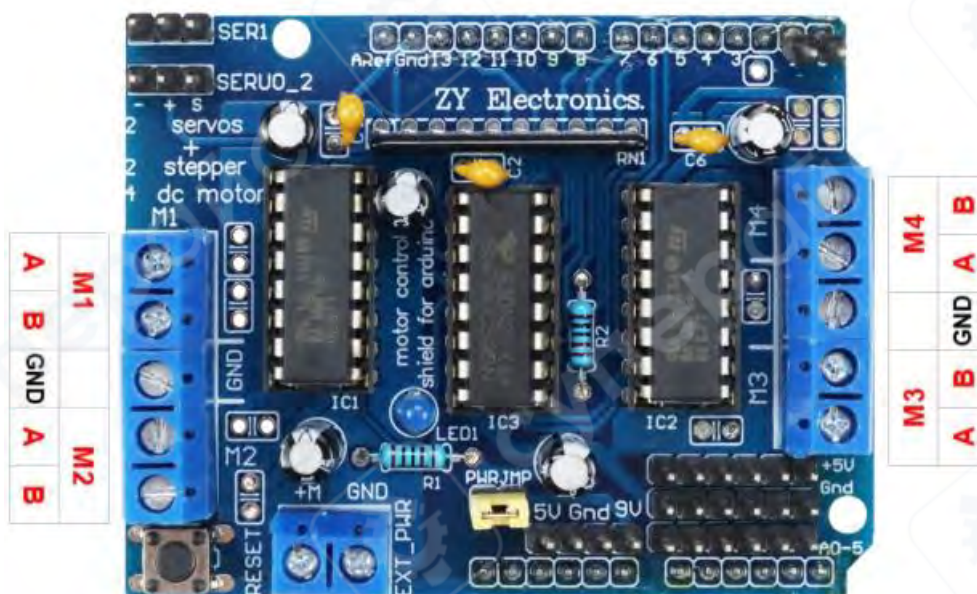
Принципы управления движения тележкой

Направления вращения моторов управляются платой 74HC595 с помощью функции shiftOut(DATA, DIR_CLK, MSBFIRST, Dir). Первые три параметра хранят базовые настройки. Последний параметр Dir — это число из десятичной системы, которое будет преобразовано в 8-битное двоичное значение. Различные значения параметра Dir определяют разные режимы движения тележки.

Известно, что каждое колесо тележки может вращаться назад и вперед, итого восемь возможных режимов поступательного движения тележки для четырех колес. При задании режима движения нужно управлять направлением вращения каждого колеса. Каждый бит в 8-битном двоичном коде соответствует прямому или обратному направлению вращения колеса. Значение бита «1» соответствует высокому уровню сигнала (колесо вращается); «0» — низкому уровню сигнала (колесо неподвижно).

На примере переменной Forward, установите значение переменной 216 (8-битное двузначное значение будет 11011000). Биты 1, 2, 4 и 5 соответствуют высокому уровню сигнала. Таким образом, согласно схеме ниже, на пины управления моторами M3-A, M4-A, M2-B и M1-B подается сигнал высокого уровня. Все колеса вращаются в прямом направлении, тележка едет вперед.

При этом, если подать сигнал высокого уровня наоборот (M3-B, M4-B, M2-A и M1-A), колеса начнут вращаться в обратном направлении, тележка поедет назад. Нельзя подавать сигнал высокого уровня на пины A и B одного мотора одновременно, так как колесо не может вращаться вперед и назад одновременно. Именно поэтому в 8-битном значении только четыре бита могут быть равны «1», а остальные четыре будут равны «0».



Чтобы узнать, какие биты 8-битного значения будут равны «1», можно использовать переменную mytest. Установите её десятичное значение 1, двоичное 8-битное будет равно 00000001.

```
22 const int R_turn = 57; //turn right
23
24 const int mytest = 1; //test
```

Добавьте в строку вызова функции Motor в теле цикла loop() параметр mytest как показано ниже (удалите параметр по окончании теста). Проследите за вращением двигателя с учетом mytest.

```
53 void loop()
54 {
55     distance = SR04(Trig,Echo); //Acquisition of ultrasonic distance
56     control_func(); //Call the Bluetooth car control function
57     Motor(mytest, Speed1, Speed1, Speed1, Speed1);
58 }
```

Ниже дана таблица соответствия значений переменной движения и направления вращения моторов. Если 8-битное значение равно 00000001, будет вращаться один мотор. Если задать десятичное 2 (00000010), будет вращаться другой мотор. При значении 4 (00000100), вращается третий мотор.

Для движения тележки вперед все моторы должны вращаться в прямом направлении, чему соответствует двоичное число 11011000, преобразованное из десятичного 216. Именно так задаются восемь различных режимов движения для четырех моторов с двумя направлениями вращения.

Помните, что два бита управления направлением вращения мотора не могут быть одновременно равны «1» (высокий уровень сигнала), иначе мотор может выйти из строя. Значения переменной управления для режимов движения даны ниже.

8-битное значение управления	1000 0000	0100 0000	0010 0000	0001 0000	0000 1000	0000 0100	0000 0010	0000 0001		
10-тичное значение управления	128	64	32	16	8	4	2	1		
Пин высокого уровня сигнала	M3-A	M4-A	M3-B	M2-B	M1-B	M1-A	M2-A	M4-B		
Позиция колеса	Правое заднее	Правое переднее	Правое заднее	Левое заднее	Левое переднее	Левое переднее	Левое заднее	Правое переднее	8-битное значение управления	10-тичное значение управления
Движение вперед	1	1	0	1	1	0	0	0	1101 1000	216
Движение назад	0	0	1	0	0	1	1	1	0010 0111	39
Смещение влево	0	1	1	1	0	1	0	0	0111 0100	116
Смещение вправо	1	0	0	0	1	0	1	1	1000 1011	139
Вращение против часовой стрелки	1	1	0	0	0	1	1	0	1100 0110	198
Вращение по часовой стрелке	0	0	1	1	1	0	0	1	00111001	57

В крайнем правом столбце показаны значение переменной Dir из функции shiftOut для задания разных режимов вращения.

Ультразвуковой дальномер

В данном разделе описан принцип работы ультразвукового модуля и программные методы измерения расстояния.

Ультразвуковой датчик

Звуковые волны генерируются механическими вибрациями и в разных средах распространяются с разными скоростями. Ультразвуковые волны характеризуются высокой интенсивностью, медленной потерей энергии и большой дальностью распространения в различных средах, благодаря чему ультразвук часто используют для измерения расстояний.



Параметры ультразвукового модуля:

Наименование модуля	Ультразвуковой модуль HC-SR04
Рабочее напряжение	DC 5 В
Рабочий ток	15 мА
Рабочая частота	40 кГц
Максимальный диапазон	4 м
Минимальный диапазон	2 см
Угол измерения	15°
Входной сигнал триггера	TTL-импульс длительностью 10 мкс
Выходной эхо-сигнал	Output TTL level signal, proportional to the range
Размеры	45 x 20 x 15

Принцип ультразвукового измерения

Поскольку в воздухе ультразвуковые волны распространяются медленно, легко зафиксировать эхо-сигналы, отражаемые структурами на пути распространения УЗ-сигнала. Благодаря высокому разрешению точность ультразвукового метода тоже очень высокая. Ультразвуковые датчики отличаются простотой конструкции, малыми габаритами и надёжностью в обработке сигналов. Ультразвуковое измерение дальности делает управление в режиме реального времени быстрым, точным и удобным, а точность измерения соответствует промышленным стандартам.

Есть множество методов ультразвукового измерения. В данной системе измеряется время распространения ультразвуковых волн от передатчика до приёмника в воздушной среде. Измеренное время умножается на скорость звука в воздухе, и так вычисляется расстояние, которое преодолела звуковая волна.

Когда ультразвуковой передатчик излучает волну в заданном направлении, микроконтроллер запускает таймер. Волна распространяется в воздухе, а затем, отразившись от препятствия на пути, тут же возвращается обратно. Ультразвуковой приёмник фиксирует отражённую волну, и микроконтроллер останавливает отсчёт времени.

Формула расстояния: $S = VT/2$

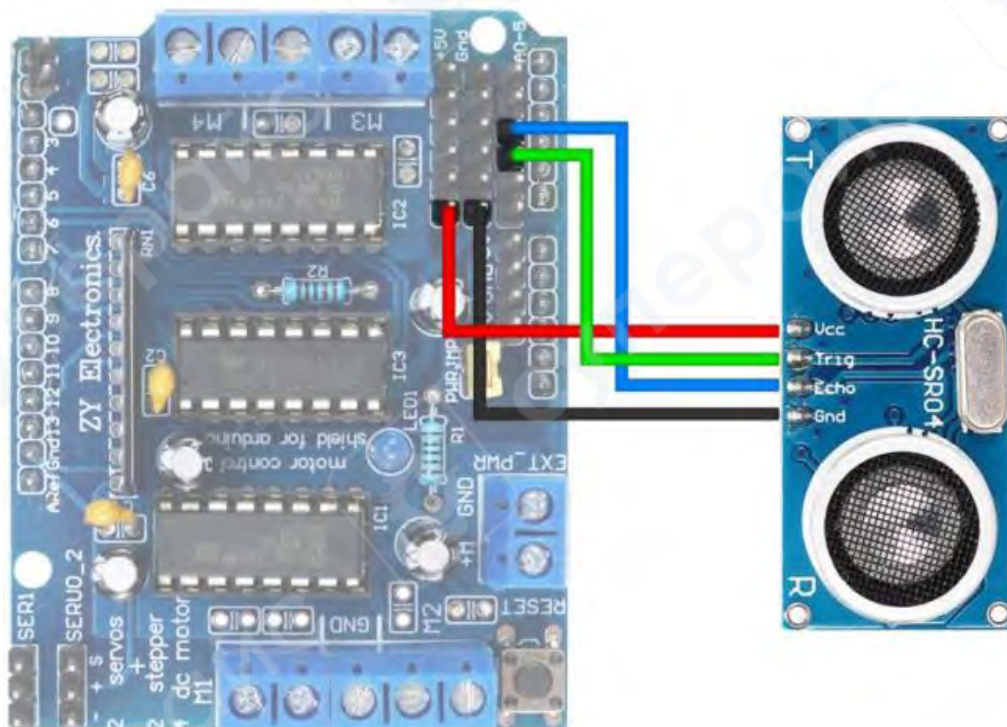
Максимальный диапазон измерения расстояния ограничен четырьмя факторами: амплитудой ультразвуковых волн, текстурой отражающей поверхности, углом между падающей и отраженной волной и чувствительностью приемника. Минимальный диапазон измерения расстояния определяются способностью приемника принимать акустический импульс.

На вход сигнала триггера (TRIG) поступает сигнал высокого уровня длительностью более 10 мкс. Когда ультразвуковой передатчик получает сигнал триггера, он испускает восемь прямоугольных импульсов частотой 40 кГц. В этот момент запускается таймер, который останавливается, когда на приемник попадает эхо-сигнал. Расстояние рассчитывается по формуле:

Расстояние = (Время подачи сигнала высокого уровня * Скорость звука)/2.

Схема подключения

Плата расширения	Ультразвуковой модуль
5V	VCC
A3	Trig
A3	Echo
GND	GND



Анализ кода

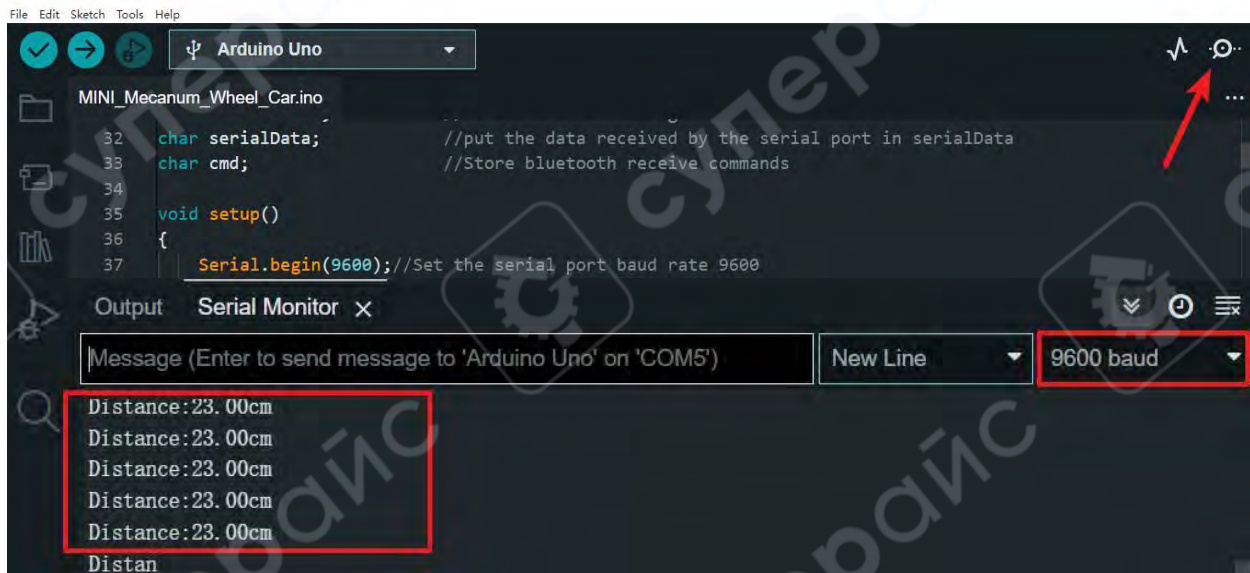
Определите пины ультразвукового передатчика и ультразвукового приемника.

```
12 //Define the pin of ultrasonic obstacle avoidance sensor
13 const int Trig = A2;      //A2 is defined as the pin Trig connected to the ultrasonic sensor
14 const int Echo = A3;      //A3 is defined as the pin Echo connected to the ultrasonic sensor
```

Функция SR04() используются для ультразвукового измерения расстояния.

```
86 int SR04(int Trig,int Echo)
87 {
88     digitalWrite(Trig,LOW);    //Trig is set to low level
89     delayMicroseconds(2);      //Wait 2 microseconds
90     digitalWrite(Trig,HIGH);   //Trig is set to high level
91     delayMicroseconds(10);     //Wait 15 microseconds
92     digitalWrite(Trig,LOW);    //Trig is set to low
93
94     float cm = pulseIn(Echo,HIGH)/58; //Convert the ranging time to CM
95     Serial.print("Distance:");   //Character Distance displayed in serial port monitor window:
96     Serial.print(cm);
97     Serial.println("cm");
98     return cm;                  //Returns cm value ranging data
99 }
```

Результаты измерения в среде Arduino IDE можно вывести на печать через COM-порт, нажав кнопку в верхнем правом углу (скорость передачи данных 9600 бод).



Функция обхода препятствий

Если расстояние, определенное ультразвуковым модулем, меньше заданного расстояния, это значит, что тележка столкнулась с препятствием после чего отъехала назад и сместилась влево, чтобы объехать препятствие.

Анализ кода

Задайте значение переменной расстояния distance.

```
31 int distance = 0;           //Variables for storing ultrasonic sensor measurements
32 char serialData;           //put the data received by the serial port in serialData
33 char cmd;                   //Store bluetooth receive commands
```

Функция обхода препятствий AvoidingObstacles().

```
166 void AvoidingObstacles()
167 {
168     if((distance > 20 ) || cmd == 'F')//If the distance is greater than 20cm or bluetooth
169     {
```

Если расстояние, измеренное ультразвуковым сигналом, больше 20 (препятствий нет), тележка движется вперед, в противном случае — назад, а затем влево, чтобы обойти препятствие. При необходимости можно изменить направление обхода, и тележка будет двигаться назад и вправо, а не назад и влево.

```
171         if(distance > 20)//Again, determine if the distance is really greater than 20cm
172         {
173             Motor(Forward,Speed1,Speed2,Speed3,Speed4); //Call forward function
174         }
175         else //Otherwise the distance is less than 20
176         {
177             Motor(Back,Speed1,Speed2,Speed3,Speed4);//retreat
178             delay(500);
179             Motor(Left,Speed1,Speed2,Speed3,Speed4);//Turn left to change the direction of the car
180             delay(500);
181         }
```


Управление через Bluetooth

В данном разделе описываются принципы работы модуля и управления через Bluetooth.

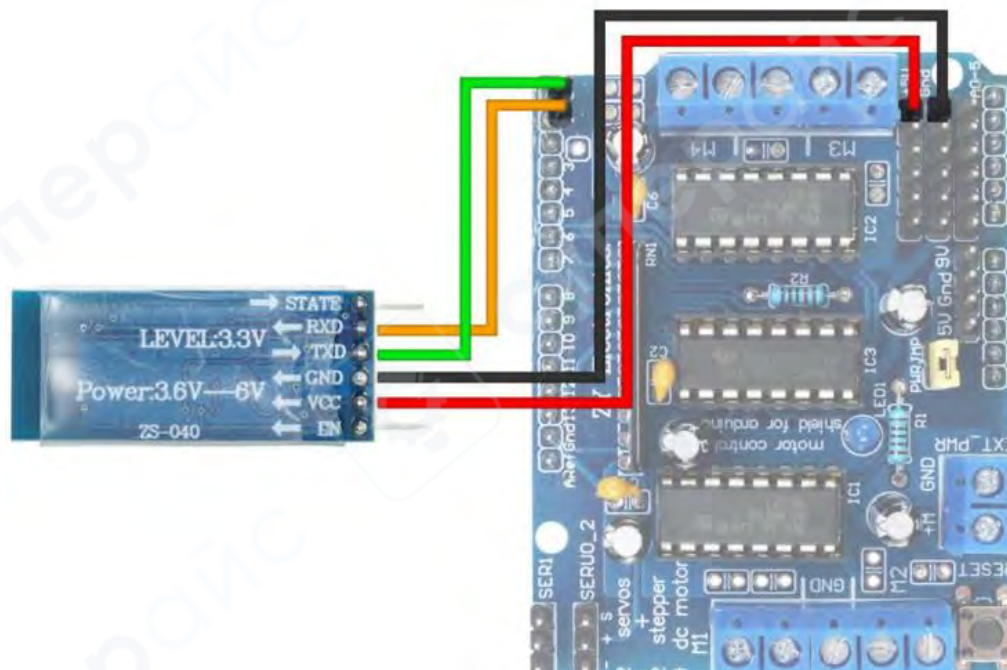
Модуль Bluetooth



Модуль Bluetooth (BT05) имеет пины RXD, TXD, GND и VCC. При подключении к главной плате порты TX и RX взаимно соединены. Порт TX платы Arduino подключается к порту RX модуля Bluetooth. Порт RX платы Arduino подключается к порту TX модуля Bluetooth.

Схема подключения

Плата расширения	Модуль Bluetooth
(TX)1	RXD
(RX)0	TXD
GND	GND
5V	VCC



Анализ кода

Функция `control_func()` осуществляет дистанционное управление через Bluetooth. Если значение данных с Bluetooth-модуля `Serial.available() > 0`, принятые данные сохраняются в переменную `serialData`.

```
106 void control_func()
107 {
108     if(Serial.available() > 0) //Determine if the received data is greater than 0
109     {
110         serialData = Serial.read(); //Receiving function
    }
```

Данные об управлении движением робота, получаемые с Bluetooth, оцениваются на основе заданных условий, а затем сохраняются в переменную `cmd`.

```
112     if ('F' == serialData ) cmd = 'F'; //If the data received by the serial port is character F, save F to CMD
113     else if('B' == serialData ) cmd = 'B'; //If the data received by the serial port is character B, save F to CMD
114     else if('L' == serialData ) cmd = 'L'; //If the serial port receives data as the character L, save F to CMD
115     else if('Y' == serialData ) cmd = 'Y'; //If the serial port receives data as the character Y, save F to CMD
116     else if('S' == serialData ) cmd = 'S'; //If the serial port receives data as character S, save F to CMD
117     else if('C' == serialData ) cmd = 'C';
118     else if('D' == serialData ) cmd = 'D';
```

Данные об управлении скоростью оцениваются похожим образом. Если скорость меньше 245, нажмите «+», чтобы увеличить значение скорости. Если скорость больше 30, нажмите «-», чтобы уменьшить значение скорости.

```
119     else if( serialData == '+' && Speed1 < 245)//If you receive a string plus, the speed increases
120     {
121         Speed1 += 10; //We're going to increase the velocity by 10 at a time
122         Speed2 = Speed1;
123         Speed3 = Speed1;
124         Speed4 = Speed1;
125     }
126     else if( serialData == '-' && Speed1 > 30)//When I receive a string -- the speed decreases
127     {
128         Speed1 -= 10; //I'm going to subtract 10 at a time
129         Speed2 = Speed1;
130         Speed3 = Speed1;
131         Speed4 = Speed1;
132     }
```

Определите значение переменной `cmd`, чтобы управлять тележкой в режимах «Вперед», «Назад», «Влево», «Вправо», «Стоп», «Поворот налево» и «Поворот направо».

```
136     if('F' == cmd) //If Bluetooth receives the string F, the dolly moves forward and enables obstacle avoidance
137     {
138         AvoidingObstacles();//
139     }
140     else if('B' == cmd) //Bluetooth receives string B, car backs up
141     {
142         Motor(Back,Speed1,Speed2,Speed3,Speed4);
143     }
144     else if('L' == cmd) //Bluetooth received the string L, car left translation
145     {
146         Motor(Left,Speed1,Speed2,Speed3,Speed4);
147     }
```

Когда значение переменной равно «F», вызывается функция `AvoidingObstacles()`, тележка движется вперед или обходит препятствия.

Когда значение переменной равно «B», вызывается функция `Motor(Back,Speed1,Speed2,Speed3,Speed4)`, тележка движется назад.

Таким образом, можно определять разные символьные значения `cmd`, чтобы задать те или иные режимы движения тележки.

Спецификация и сборка

Комплектующие



Болт с круглой головкой M1.6
(4 шт.)



Гайка M1.6 (4 шт.)



Болт с плоской головкой
M3.8 (14 шт.)



Болт с плоской головкой M3.12
(14 шт.)



Болт с круглой головкой M3.30
(8 шт.)



Гайка M3 (24 шт.)



Медная стойка M3.15
(4 шт.)



Медная стойка M3.30
(2 шт.)



Крепежная скоба для
моторов (4 шт.)



Акриловая
платформа



Муфты и болты M3x20 (4
шт.)



Отвертка



Платформа для U3-модуля



Моторы (4 шт.)



Колеса (4 шт.)



Плата разработки



Плата расширения



Ультразвуковой модуль



Модуль Bluetooth



Блок батареек



USB-кабель Type-C



Кабель Dupont «мать-мать»
20 см 10P

Установка медных стоек



- ① Медная стойка
M3x30 (2 шт.)



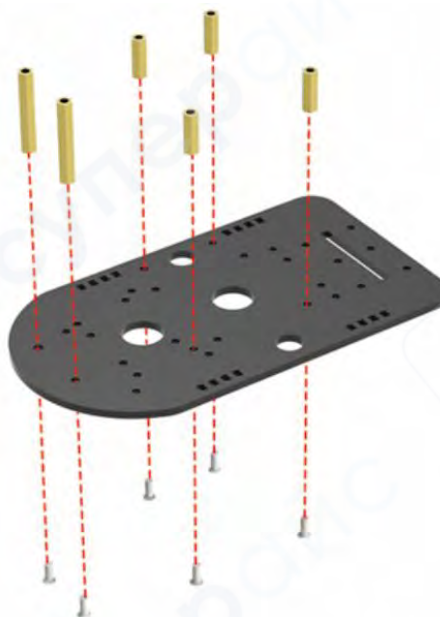
- ② Медная стойка
M3x15 (4 шт.)



- ③ Болт с плоской головкой
M3x8 (6 шт.)



- ④ Акриловая платформа



Вставьте медные стойки в отверстия на лицевой стороне акриловой платформы, как показано выше. Будьте внимательны, не перепутайте лицевую и обратную стороны платформы.

Установка крепежных скоб



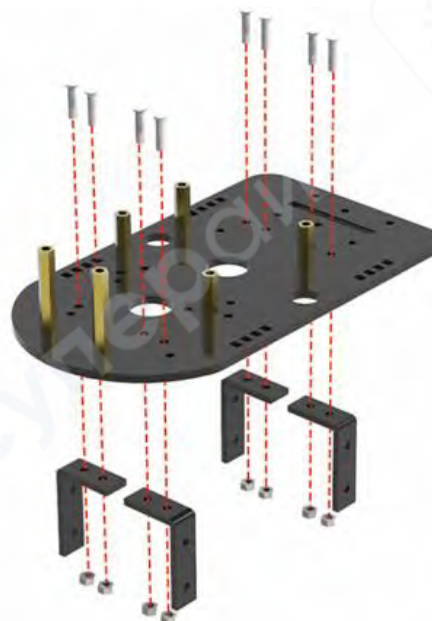
- ① Болт с плоской головкой
M3x12 (8 шт.)



- ② Гайка M3 (8 шт.)



- ③ Крепежная скоба
для моторов (4 шт.)



Прикрепите крепежные скобы к обратной стороне платформы, как показано выше.

Установка моторов

- ① Болт с плоской головкой
M3x30 (8 шт.)

- ② Гайка M3 (8 шт.)

- ③ Моторы (4 шт.)



Закрепите четыре мотора на скобах.

Конец мотора с припоем должен смотреть наружу и
конец с выступом должен смотреть внутрь.

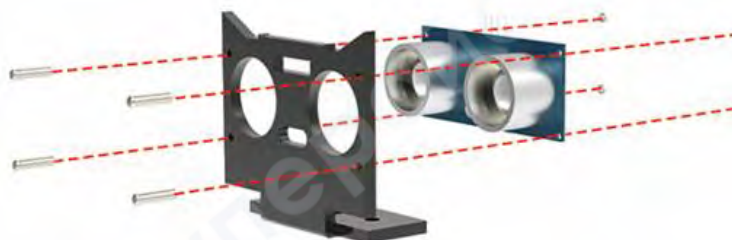
Установка ультразвукового модуля

- ① Болт с круглой головкой
M1.6 (4 шт.)

- ② Гайка M1.6 (4 шт.)

- ③ Ультразвуковой модуль

- ④ Платформа
ультразвукового модуля



Установите ультразвуковой модуль в
соответствующую платформу, как показано выше.

- 
- ① Болт с плоской головкой
М3.8 (2 шт.)

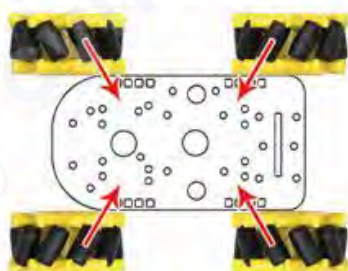


Закрепите платформу с ультразвуковым модулем на медные стойки, как показано выше.

Установка колес

- 
- ① Муфты и болты М3х20
(4 шт.)

- 
- ② Колеса (4 шт.)



Сперва установите муфту на вал мотора, затем закрепите колесо на муфте.

Будьте внимательны, соблюдайте ориентацию правых и левых колес, как показано выше.

Установка платы разработки и платы расширения

- ① Болты с плоской головкой
M3x8 (4 шт.)



- ② Плата разработки



- ③ Плата расширения



Зафиксируйте плату разработки на медных стойках, затем установите плату расширения на плату разработки.

Будьте внимательны, соблюдайте правильное положение платы разработки и платы расширения.

Установка блока батареек

① Болты с плоской головкой М3х12 (4 шт.)

② Гайка М3 (4 шт.)

③ Блок с батарейками



Сперва зафиксируйте блок батареек на акриловой платформе, затем вставьте батарейки в блок, соблюдая полярность. Подключите кабель питания к плате разработки.

Батарейки в комплект не входят. Подготовьте батарейки номиналом 3.7 В 18650 самостоятельно.

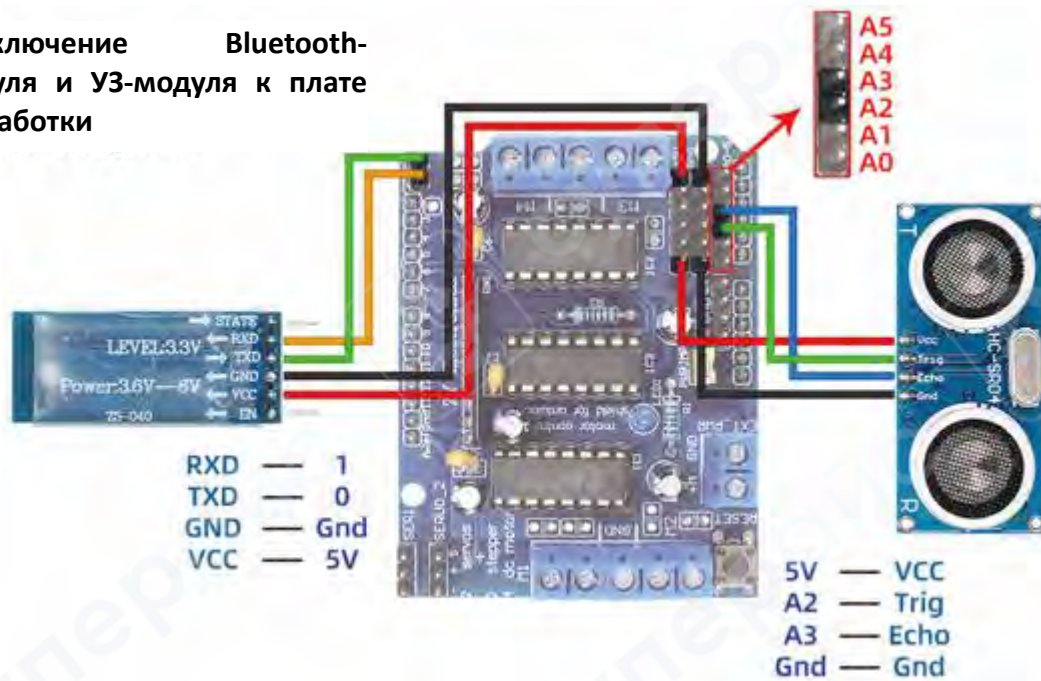
Установка Bluetooth-модуля

① Bluetooth-модуль



Подключите Bluetooth-модуль к плате разработки, как показано в разделе выше. Установите Bluetooth-модуль на плату разработки.

Подключение Bluetooth-модуля и УЗ-модуля к плате разработки



Схемы подключения

Схема подключения моторов:

Соблюдайте правильное положение модуля относительно тележки

