

Курс «Основы робототехники». Занятие 4

Управление роботами по Bluetooth

“Все становится лучше, когда в нем есть Bluetooth”

Шелдон Купер, Теория Большого Взрыва

Библиотека SoftwareSerial

Используется для связи Arduino с другими устройствами. Будем использовать следующие функции:

#include <SoftwareSerial.h> – подключение библиотеки SoftwareSerial

SoftwareSerial mySerial(11, 12) – создание объект mySerial, осуществляющего связь, в данном случае через пины 11 и 12

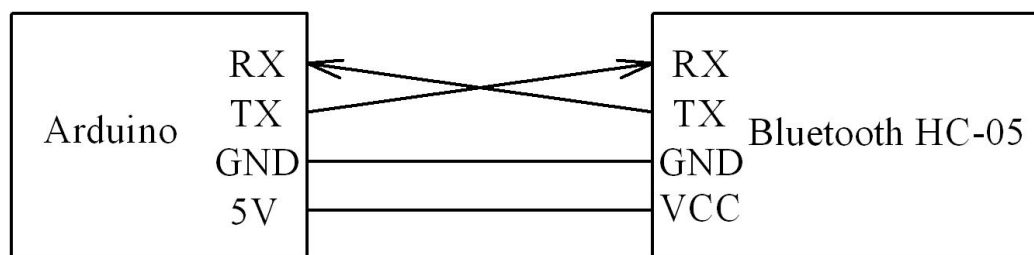
mySerial.begin(9600) – начать связь, скорость 9600 бит в секунду. Без выполнения этой команды в блоке *setup* все остальные команды работать не будут. 9600 – стандартная скорость для связи с компьютером и используемыми Bluetooth модулями. Если внешнее устройство работает на другой скорости (например, 57600) – необходимо указать эту скорость, иначе связь не будет установлена

mySerial.available() – проверка, принят ли символ

command = mySerial.read() – запись принятого символа в символьную (*char*) переменную *command*

Bluetooth модуль

Вставляется в макетную плату робота. Подключение модуля показано на схеме:



RX – receiver, приемник (получает информацию)

TX – transmitter, передатчик (передает информацию)

При этом контакты должны быть включены крест-накрест: RX одного устройства к TX другого, TX первого устройства к RX второго.

Пример кода для связи:

```
#include <SoftwareSerial.h>           // подключаем библиотеку SoftwareSerial
```

```

SoftwareSerial mySerial(11, 12);           // создаем объект mySerial – связь с Bluetooth через
пины 11 и 12
char command;                             // переменная для хранения принятого символа

void setup() {
    // подключаемся к Bluetooth модулю, соединенному с Arduino по Software Serial:
    mySerial.begin(9600);
    // тут настройте пины, к которым подключен драйвер Вашего робота, как Output
}

```

Для управления роботами на Arduino с планшета или смартфона имеется ряд приложений: как текстовые, передающие и принимающие буквы и числа на робота и с него (например, Bluetooth SPP), так и графические, с готовыми или создаваемыми пользователем кнопками, нажатие на каждую из которых передает на робота по Bluetooth какую-либо букву или цифру. Для получения этого символа на Arduino в блоке loop используется следующий код:

```

if (mySerial.available()) {                // если есть принятый символ,
    command = mySerial.read();             // то читаем его и сохраняем в command
}

```

Для управления роботами можно использовать приложение **Arduino Bluetooth RC Car** (см. приложенный файл о его установке и подключении к роботу по Bluetooth). Данное приложение передает по Bluetooth символы F (Forward, вперед), B (Backward, назад), R (Right, вправо), L (Left, влево) и многие другие, но для начала используем только перечисленные.

Напишите программу, включающую моторы робота в зависимости от пришедшего символа. Можно использовать несколько if, но, учитывая, что в дальнейшем будут отрабатываться и остальные кнопки приложения, передающие свои символы, более корректно будет использовать функцию switch-case:

```

switch (command) {                        // выполняем команду в соответствии с принятым символом
    (если он есть)
    case 'F': // принят символ с кнопки Вперед
        // команды для езды вперед

```

```

break;

case 'B':    // принят символ с кнопки Назад
    // команды для езды назад
    break;

case 'L':    // принят символ с кнопки Влево
    // команды для езды влево
    break;

case 'R':    // принят символ с кнопки Вправо
    // команды для езды вправо
    break;

default:    // если принят какой-то другой символ – остановиться
    // команды для остановки
    break;
}

```

Практическое занятие 1: напишите программу, реализующую движение робота по управлению с Bluetooth используя приведенные выше части кода. **Вместо закомментированных частей вида «команды для езды назад» вставьте соответствующие команды (4 строки analogWrite).** При этом для того, чтобы робот ехал прямо, используйте разные скорости для правого и левого мотора, найденные на прошлом занятии.

Практическое занятие 2: Добавим управление скоростью робота по кнопкам фар вверху слева в приложении Arduino Bluetooth RC Car. Эти кнопки по нажатиям отправляют

- кнопка  – символы W и w
- кнопка  – символы U и u

```

case 'U':
    speed = speed + 10;

```

```

break;
case 'u':
    speed = speed + 10;
    break;

case 'W':
    speed = speed - 10;
    break;
case 'w':
    speed = speed - 10;
    break;

```

Практическое занятие 3: добавим обработку диагональных направлений, чтобы робот мог не только разворачиваться на месте, но и, как автомобиль, двигаться вперед-вправо, вперед-влево и назад-вправо, назад-влево. Символы, которые отправляются при одновременном нажатии на такие кнопки, приведены в приложении, но для упрощения работы была сделана таблица:

G – вперед влево ☐	F – вперед ☐	I – вперед вправо ☐
L – влево ☐		R – вправо ☐
H – назад влево ☐	B – назад ☐	J – назад вправо ☐

Соответственно, добавьте в программу управления следующий код:

```

case 'G ':    // принят символ с кнопок Вперед и Влево
    // команды для езды вперед влево
    break;
case 'I ':    // принят символ с кнопок Вперед и Вправо
    // команды для езды вперед вправо
    break;
case 'H ':    // принят символ с кнопок Назад и Влево
    // команды для езды назад влево
    break;
case 'J ':    // принят символ с кнопок Назад и Вправо
    // команды для езды назад вправо

```

```
break;
```

Примечание: для движений «по диагонали» нужно на один мотор подавать меньшую скорость, чем на другой. Например, для езды вперед влево надо дать обоим моторам команду ехать вперед, но на левый мотор подать скорость поменьше, чтобы робот при езде забирал влево.