РОБОРОВЕР М1

“Описание программного интерфейса по программированию робота

РобоРовер М1 4WD”

(MR-RT-RR2-M1-02-4WD)

Для программы

RR2-M1-02-4WD-CURIE101-OA1-3-stable

***ВНИМАНИЕ!***

***Прежде чем начать использовать робота РобоРовер М1 внимательно ознакомьтесь с настоящим описанием.***

Maxxlife Robot® & МРобот®

2016 Revision 1.0

**От авторов.**

Спасибо за то, что Вы используете наш продукт РобоРовер М1.

Вас приветствует команда разработчиков роботов и мехатронных устройств Maxxlife Robot® и интернет-магазин МРобот (mrobot.by)!

Наша команда обеспечивает все условия для развития сообщества по робототехнике и “сделай сам” устройствам. Наша цель – это предоставить любому желающему возможность воплотить свои мечты в реальность.

Специально для этого мы создали робототехническую платформу РобоРовер М1. На ее основе можно создавать колесных роботов различного типа и назначения. Это образовательный комплект по робототехнике, к которому и прилагается эта инструкция, чтобы Вы могли творить с удовольствием!

Предлагаю перейти “ты”, так будет проще взаимодействовать с Вами.

**Авторское право**

“Описание программного интерфейса по программированию робота РобоРовер М1 4WD”

от Maxxlife Robot® и МРобот.

Copyright © 2016 Максим Массальский

Мои контакты:

VK.com/maxxlife

FB.com/teammaxxlife

maxxliferobot@gmail.com

Оглавление

[Введение 5](#_Toc461836994)

[1. Описание переменных программы RR2-M1-02-4WD-CURIE101-OA1 6](#_Toc461836995)

[2. Описание функций программы RR2-M1-02-4WD-CURIE101-OA1 7](#_Toc461836996)

[2.1 Функции по измерению расстояний 7](#_Toc461836997)

[2.1.1 Функция для ИК-датчиков расстояния 7](#_Toc461836999)

[2.1.2 Функция для УЗ-датчика расстояния 8](#_Toc461837000)

[2.2 Функции по передвижению робота 8](#_Toc461837001)

[2.2.1 Функция по движению вперед 9](#_Toc461837002)

[2.2.2 Функция по движения назад 9](#_Toc461837003)

[2.2.3 Функции по движению влево 9](#_Toc461837004)

[2.2.4 Функции по движению вправо 10](#_Toc461837005)

[2.2.5 Функция определения столкновения 11](#_Toc461837006)

[3. Описание простейшей программы по отладке робота и программы по движению робота 11](#_Toc461837007)

[3.1 Отладочная программа 11](#_Toc461837008)

[3.2 Гоночная программа по огибанию роботом препятствий 12](#_Toc461837009)

# Введение

В данном руководстве описаны функции и переменные для программирования робота.

Для программирования робота РобоРовем М1 необходимо скачать среду программирования Arduino IDE по ссылке <https://www.arduino.cc/en/Main/Software>

Также необходимо установить в среде Arduino IDE поддержку платы Arduino 101/Genuino 101 и выбрать ее в качестве основной платы разработки в среде Arduino IDE.

В роботе установлены 2 инфракрасных датчика (ИК-датчика) расстояния Sharp GP2Y0A21YK0F и 1 ультразвуковой датчик (УЗ-датчик) расстояния HC-SR04.

В роботе установлены 4 электродвигателя типа DAGU 1:48. Для управления электродвигателями используется драйвер моторов Motor Shield L298N.

Datasheet на датчик Sharp GP2Y0A21YK0F доступен по ссылке <http://www.sharpsma.com/webfm_send/1489>

Datasheet на датчик HC-SR04 доступен по ссылке <http://www.micropik.com/PDF/HCSR04.pdf>

Описание на драйвер моторов L298N доступно по ссылке <http://www.instructables.com/id/Arduino-Modules-L298N-Dual-H-Bridge-Motor-Controll/>

# Описание переменных программы RR2-M1-02-4WD-CURIE101-OA1

Здесь будут описаны только основные переменные, которые необходимо точно знать.

**ИК-датчики**

Переменные по хранению расстояний с переднего левого и правого ИК-датчиков. Тип volatile int.

lfront

rfront

В эти переменные автоматически записываются значения расстояния с правого и левого ИК-датчиков в сантиметрах при вызове функции irMeasure(). Запрещается записывать значения в эти переменные самостоятельно, т.к. это может приводить к сбою в программе.

**УЗ-датчик**

Переменная по хранению расстояния с УЗ-датчика. Тип volatile int.

usRange

В эту переменную автоматически записывается значение расстояния с переднего УЗ-датчика в сантиметрах при вызове функции usMeasure().Запрещается записывать значение в эту переменную самостоятельно, т.к. это может приводить к сбою в программе.

# Описание функций программы RR2-M1-02-4WD-CURIE101-OA1

## Функции по измерению расстояний

# В роботе установлены 2 инфракрасных датчика (ИК-датчика) расстояния Sharp GP2Y0A21YK0F и 1 ультразвуковой датчик (УЗ-датчик) расстояния HC-SR04

**Для работы с данными датчиками написаны следующие функции:**

### Функция для ИК-датчиков расстояния

irMeasure(); -функция по получению и обновлению данных с ИК-датчиков робота. Тип void.

Вызвав эту функцию, обновляются переменные rFront и lFront. В переменных rFront и lFront хранится измеренное расстояние в сантиметрах от правого ИК-датчика и левого ИК-датчика соответственно.

Максимальный диапазон измерения расстояния ИК-датчиками от 6см до 110см. Диапазон в котором датчики получают самые точные показания без ошибок от 8см до 60см с точностью в 2см.

### Функция для УЗ-датчика расстояния

usMeasure(); -функция по получению и обновлению данных с ультразвукового датчика робота. Вызвав эту функцию обновляется переменная usRange в которой хранится значение измеренного расстояния с УЗ-датчика в сантиметрах. Тип void.

Диапазон измерения расстояния УЗ-датчика от 2см до 250см. УЗ-датчик может показывать нереалистичные значения при измерении расстояния от искривленных предметов, одежды, руки и т.д. Поэтому УЗ-датчик не всегда дает точные показания. Необходимы тесты, чтобы точно знать какие значения выдает функция.

## Функции по передвижению робота

Робот РобоРовер М1 4WD имеет четыре электродвигателя и полный привод на все 4 колеса. Двигатели управляются при помощи драйвера двигателей L298N, который управляет моторами через H-мост.

Правые 2 двигателя объединены между собой и работают в паре. Левые 2 двигателя объединены между собой и работают в паре. Робот поворачивает за счет разности угловых скоростей на левых и правых двигателях.

Чтобы роботу повернуть влево, роботу надо увеличить скорость вращения правых колес и уменьшить скорость вращения левых колес. Чтобы сделать резкий поворот влево, надо увеличить скорость вращения правых колес, а левые колеса, надо начать вращать в обратную сторону. Для первого случая есть функции go\_RightHalf(); и go\_LeftHalf(); а для второго случая есть функции go\_RightFull и go\_LeftFull();

Моторы могут изменять скорость вращения через ШИМ-сигнал, который может изменяться от 0 до 255. 255 Это максимальная скорость вращения. 0-это минимальная скорость вращения, мотор не вращается.

**Чтобы упростить работу с драйвером L298N написаны следующие функции:**

### Функция по движению вперед

go\_Forward(220,220); -функция по движению робота вперед. Вызвав данную функцию робот будет двигаться вперед. Тип int. Принимает данные о значении скорости вращения двигателей.

go\_Forward(скорость вращения левых двигателей, скорость вращения правых двигателей);

Первое значение меняет скорость вращения на левых двигателях. Второе значение меняет скорость вращения на правых двигателях.

Меняя числа (220,220) на другие, например, (250,250) робот будет двигаться быстрее. И наоборот, (160,160) робот будет двигаться медленнее. Оптимальное значение скорости для движения вперед (220,220), (180,180).

Если робот при движении прямо и при одинаковых значениях на левых и правых двигателях немного поворачивает в какую-либо сторону, то необходимо увеличить или уменьшить скорость вращения левых или правых двигателях. Например, робот при вызове функции go\_Forward(220,220)едет прямо, но немного уходит влево. Тогда надо увеличить скорость вращения на левых двигателях и поменять значения функции приблизительно так go\_Forward(230,220). Значения необходимо устанавливать экспериментальным путем.

### Функция по движения назад

go\_Backward(220,220); -функция по движению робота назад. Вызвав данную функцию робот будет двигаться назад. Тип int. Принимает данные о значении скорости вращения двигателей.

go\_Backward(скорость вращения левых двигателей, скорость вращения правых двигателей);

Первое значение меняет скорость вращения на левых двигателях назад. Второе значение меняет скорость вращения на правых двигателях назад.

### Функции по движению влево

go\_LeftHalf(100,220); **-**функция по плавному движению робота влево. Тип int. Функция принимает данные о значении скорости вращения двигателей.

go\_LeftHalf(скорость вращения левых двигателей назад, скорость вращения правых двигателей вперед);

Вызвав данную функцию, робот будет плавно поворачивать влево. Левые колеса движутся вперед, но медленнее, имеют значение 100. Правые колеса движутся быстрее вперед, имеют значение 220. Но левые и правые колеса движутся в одну сторону -вперед.

Можно вообще не вращать левые колеса, написать значение go\_LeftHalf(0,220); так робот будет поворачивать влево еще лучше.

go\_LeftFull(100,220); **-**функция по резкому движению робота влево. Тип int. Функция принимает данные о значении скорости вращения двигателей. Вызвав данную функцию, робот будет резко поворачивать влево. Левые колеса вращаются назад на скорости 100, а правые колеса вращаются вперед на скорости 220. Левые и правые колеса движутся в разные стороны.

### Функции по движению вправо

go\_RightHalf(220,100); **-**функция по плавному движению робота вправо. Тип int. Функция принимает данные о значении скорости вращения двигателей.

go\_RightHalf(скорость вращения левых двигателей назад, скорость вращения правых двигателей вперед);

Вызвав данную функцию, робот будет плавно поворачивать вправо. Правые колеса движутся вперед, но медленнее, имеют значение 100. Левые колеса движутся быстрее вперед, имеют значение 220. Левые и правые колеса движутся в одну сторону -вперед. Можно вообще не вращать правые колеса, написать значение go\_RightHalf(220,0); так робот будет поворачивать вправо еще лучше.

go\_RightFull(220, 100); **-**функция по резкому движению робота вправо. Тип int. Функция принимает данные о значении скорости вращения двигателей. Вызвав данную функцию, робот будет резко поворачивать вправо. Правые колеса вращаются назад на скорости 100, а левые колеса вращаются вперед на скорости 220. Левые и правые колеса движутся в разные стороны.

**Выбор типа функции зависит от того, как сильно необходимо повернуть роботу.**

### Функция определения столкновения

Когда на обоих ИК-датчиках расстояния становятся меньше определенного предела, то можно считать, что робот столкнулся со стеной. После этого роботу как-то надо отъехать назад и вернуться на прежний курс. Специально для этого была создана функция

irFrontGoBack();

При ее вызове робот будет двигаться назад от стены, после чего вернется на прежний курс и продолжит движение.

# Описание простейшей программы по отладке робота и программы по движению робота

Все программы доступны на github по ссылке <https://github.com/mrobotby/RoboRover_M1_4WD_101_Start>

## Отладочная программа

Для того, чтобы новичку было просто начать разбираться с роботом и производить отладку показаний с датчиков создана программа

RR2-M1-02-CURIE101-TEST1-1-stable.ino

Вгрузив данную программу в робота и открыв монитор последовательного порта, можно видеть все показания с датчиков. При неисправности, по программе можно легко определить какой датчик показывает неверные значения.

## Гоночная программа по огибанию роботом препятствий

Специально для того, чтобы сократить время на написание кода и помочь новичку была создана программа

RR2-M1-02-4WD-CURIE101-OA1-3-stable.ino

Данная программа позволяет роботу двигаться по трассе и огибать препятствия. В программе огромное количество комментариев, которые упрощают знакомство с кодом.

Эта базовая программа позволяет роботу двигаться, огибать препятствия и выезжать из тупика. Для начала работы, рекомендуется использовать ее.

Все манипуляции с кодом необходимо проводить только в бесконечном цикле void loop(). Там находится основной алгоритм робота. Для начала рекомендуется изменять параметры в бесконечном цикле void loop().

Ниже будет приведен пример основного кода из void loop()

1. if ((lFront > 50) && (rFront > 50))
2. {
3. // digitalWrite(13,HIGH);
4. go\_Forward(220,220);
5. }
6. //В другом случае
7. else
8. {
9. //Если с левого и правого ИК-датчика расстояние маленькое и с УЗ-датчика маленькое
10. if ((lFront <= 30) && (rFront <= 30) || (usRange <= 15))
11. {
12. //То думаем, что робот рядом со стеной
13. //Вызываем функцию по движению робота назад и выезду из тупика
14. irFrontGoBack();
15. }
16. //В другом случае
17. else
18. {
19. //Если расстояние с правого ИК-датчика меньше чем с левого ИК-датчика
20. if ((rFront <= lFront))
21. {
22. //То робот поворачивает влево
23. go\_LeftFull(100,180);
24. }
25. //Если расстояние с левого ИК-датчика меньше чем с правого ИК-датчика
26. if (lFront < rFront)
27. {
28. //То робот поворачивает вправо
29. go\_RightFull(180,100);
30. }
31. }
32. }

**Для начала рекомендуется подстроить данную программу под трассу, используя следующую последовательность шагов:**

1. Необходимо настроить значения расстояния при которых робот начинает ехать прямо в строке 1. Правильно настроенное расстояние будет позволять роботу на прямых ехать прямо, без дерганий из стороны в сторону. Слишком маленькое расстояние может привести к тому, что робот начнет слишком поздно поворачивать, поэтому необходимы эксперименты.
2. Необходимо настроить скорость движения прямо в строке 4, учитывая тот факт, что когда расстояние становится меньше определенного предела, роботу надо еще успеть повернуть, а чем больше была скорость движения прямо, тем быстрее надо повернуть роботу! Слишком большая скорость движения прямо может привести к тому, что робот просто не успеет войти в поворот и столкнется со стеной!
3. Необходимо настроить значения расстояния с ИК-датчиков, в том случае, когда роботу надо ехать назад в строке 10. Выяснить экспериментальным путем, какие значения расстояния выдают датчики при столкновении со стеной. Правильно настроенный параметр, будет определять только столкновение со стеной без ложных срабатываний.
4. В строке 23 и 29 необходимо настроить силу поворота робота влево и вправо.
5. Когда робота более-менее умеет двигаться по трассе, сохраним эту программу и можно начинать свободное творчество☺

P.S. Вот такое небольшое описание, оно будет только пополняться и становится все лучше и лучше вместе в вами! Успехов!