RAcinator’s User Manual

RACINATOR INC.

Table of Contents

[Table of Figures 3](#_Toc258999831)

[1.0 INTRODUCTION 4](#_Toc258999832)

[2.0 GETTING STARTED 4](#_Toc258999833)

[2.1 What Is Inside? 4](#_Toc258999834)

[2.2 How to Use It? 4](#_Toc258999835)

[3.0 USING SOFTWARE 6](#_Toc258999836)

[3.1 Software Installation guideline 6](#_Toc258999837)

[3.2 Software Function 6](#_Toc258999838)

[3.3 Software modification 8](#_Toc258999839)

[3.3.1 Driving functions 8](#_Toc258999840)

[3.3.2 Turning 9](#_Toc258999841)

[4. 0 TROUBLESHOOTING 11](#_Toc258999842)

[4.1 Hardware Trouble Shooting 11](#_Toc258999843)

[4.2 Software Trouble Shooting 13](#_Toc258999844)

[5.0 MAINTENANCE 14](#_Toc258999845)

[5.1 Charging and Replacing the Battery 14](#_Toc258999846)

[5.2 Cleaning the S.V.R 14](#_Toc258999847)

[5.3 In general 14](#_Toc258999848)

[5.3 Warning 15](#_Toc258999849)

[INDEX 16](#_Toc258999850)

# Table of Figures

Figure 1 Switch 5

Figure 2 Drivinig 9

Figure 3 Turning 10

# 1.0 INTRODUCTION

Thank you for purchasing the Racinator. Most of all the features and functionality is written in this user manual. This manual can be a useful tool to understand the entire function of the robot and should be kept with the robot for quick reference. Before using your robot, please read the manual carefully.

# 2.0 GETTING STARTED

## 2.1 What Is Inside?

* Pre-assembled version of the robot
* 1 user manual
* 1 technical manual
* 1 CD software installer
* 1 ISP header
* 1 LPT-ISP cable

## 2.2 How to Use It?

Step 1: Power On

* Open the battery pack
* Insert six AA batteries in the battery pack.
* Each one has to be inserted in the specific cap following the positive and negative side.

Step 2: Place the robot on a track

* Make sure you are running your robot on a flat and smooth surface
* Make sure that the wheels, the battery pack and the all other connections are well connected to the main body of the robot
* Place your robot in a START position

Step 3: Make the robot to follow a black line

* Turn on the robot by putting the on/off switch to the ‘ON’ position (refer figure 1)

Step 4: Power Down

* Push the switch to the ‘OFF’ position (refer to Figure 1)



Figure Switch

# 3.0 USING SOFTWARE

## 3.1 Software Installation guideline

Install the following programs in order:

1) AVR Studio

This program is used to write, modify, compile, and debug for a wide range of microcontroller, software based on C programming.

2) PonyProg

This program is used to upload software to the microcontroller.

## 3.2 Software Function

Our program contains eight files:

Code.c, Driving.c, Initializations.c, InitializationPWM.c, Line\_Obj.c, Turning.c, TimeDelay.c, Cases.c, Global.h

* Code.c is where the main commands are executed; the usual C main function is placed there. Thus, we call all the functions from here most of the time. First we initialize the robot, by setting the right ports. We call functions from Initializations.c for that. Also, the motors and the H-bridge ports are set. The PWM set-up function is called from InitializationPWM.c. After the initializations, a line search is done. A function is called to see if the roboit is on a line, i.e. we check if our side sensors are aligned to a line. When we find a line we record into a variable if the right or left side of the robot is sensing that line. That way we know if we are on the right or left side of the track. After that, our program basically retrieved the conditions and set the appropriate commands by calling the functions from our included files.
* Initializations.c enables the ports for the H-bridge to communicate with our microcontroller.
* InitializationPWM.c contains the setting of the timer registers for the pwm signals to be active. There is a function there when called with the right parameter will change the duty cycles to the motors.
* Global.h has the global variables that should be accessed by our entire program.
* Timedelay.c functions allow us to delay commands we think appropriate for a certain cases.
* Turning.c make us turn with various angle, the latter being the passing parameters to the left and right functions.
* Driving.c has the functions that make the robot go forward and backward by enabling the appropriate ports of the H-bridge. We modify the speed by varying the PWM.
* Line\_Obj.c monitors the robot moving along the track lines. That’s where we instruct the robot to get on the track after observing the data from the sensors. We try, there, to accurately give the right commands so that the robot always follows the left or right line.
* Cases.c is where the logic of the program stands. All the possible cases or situations are examined there. By fetching the sensors’ inputs we construct some kind of logic table. This will help to make decisions in the Line\_Obj.

## 3.3 Software modification

### 3.3.1 Driving functions

One may modify the speed of the Racinator by setting a higher “speed” value in the driving functions: driveForward (speed), speed can be set between values 75 and 150, where the higher the value, the faster the car will move. Speed ranges outside these values may cause problems.

The speed values will set the duty cycle using the “setPWMOuput()” function

* driveForward(uint8\_t speed)
* driveBackward(uint8\_t speed)
* setPWMOutputLM(value)
* setPWMOutputRM(value)

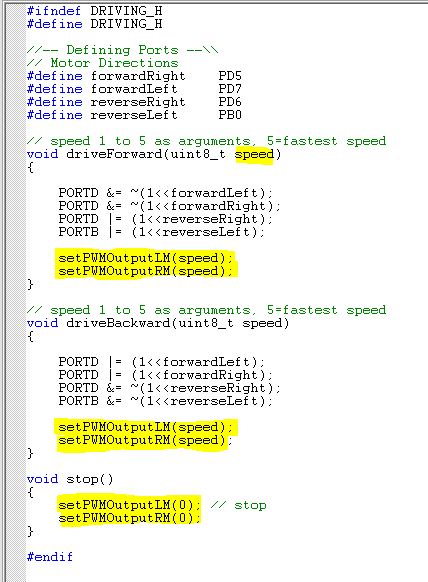


Figure Drivinig

### 3.3.2 Turning

One may modify the duration that the Racinator will turn by modifying a “time” value in the turning functions: turnRight(time) and turnLeft(time). The pre set duration are named: lowest, low, high and highest. The higher duration gives a bigger turn. Pre set time is set to five degree and it goes up by five as time increases by one.

* turnRight(uint8\_t time)
* turnLeft(uint8\_t time)

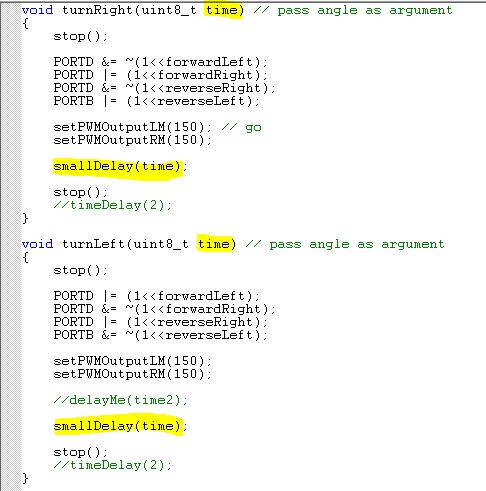


Figure Turning

# 4. 0 TROUBLESHOOTING

In case some minor problems are encountered, here are some solutions you can follow in order to make your robot work again.

## 4.1 Hardware Trouble Shooting

|  |  |  |
| --- | --- | --- |
| Problems | Causes | Solutions |
| The power of the robot does not turn on | Batteries are not inserted | Make sure the battery is inserted. If you do not, please insert the battery |
| Batteries are not inserted correctly | Check the battery sign and reinsert the battery |
| Battery power is low | Insert a new battery or recharge it |
| Battery pack is not properly connected to the breadboard | Re-wire the connection as given electrical schematics in the technical manual or call us |
| The robot does not move on the floor | Switch is not placed in “on” position | Place the switch in “on” position |
| The wheels do not rotate | Motor and wheel are not properly connected | Check to see if the motor control wires are connected properly between the breadboard and motor. If not, please call us for help. |
| Software Issue | Contact us |
| The robot does not follow the black electrical tape | Dirt disturbs | Wipe off any dirt on the sensors |
| Sensors are defected | Call us for a replacement kit |
| The robot runs at dark environment | The robot might not work at dark environment. Try to test in light environment |
| The robot does not detect obstacles | Dirt disturbs sensors | Wipe off any dirt on the sensors |
| Sensors are defected | Call us for a replacement kit |
| The robot runs at dark environment | The robot might not work at dark environment. Try to test in light environment |

## 4.2 Software Trouble Shooting

|  |  |  |
| --- | --- | --- |
| Problems | Causes | Solutions |
| Modified programs cannot be uploaded | USB/ISP is not connected properly | Make sure they are connected to the right ports. |
| Atmega8 Microcontroller is defected | Replace Atmega8 Microcontroller with a new chip and try again to upload programs |
| Software crashes | Unknown | Try to reboot your computer |
| Memory overflow | Atmega8 Microcontroller has limited amount of memory | Reset the memory |

# 5.0 MAINTENANCE

## 5.1 Charging and Replacing the Battery

* If your robot battery level becomes low, the robot will automatically shuts off or move slow.
* Use approved charger
* If the batteries begin to leak or corrode, turn off the robot immediately and take them off from the battery pack.
* Only use AA size battery: Do not overload the robot with a higher voltage battery.

## 5.2 Cleaning the S.V.R

* Use only air duster to clean up dirt

## 5.3 In general

* Always take out the batteries when the robot is not in use
* Make sure that ON/OFF switch is placed on the “OFF” position when the robot is not in use
* Keep the robot in a dry and cool environment

## 5.3 Warning

* **KEEP OUT OF ANY MAGNETIC MATERIALS**
* **DO NOT** run the robot in a unsuitable area.
* **DO NOT** keep in a humid place
* **DO NOT** use the wrong battery charger
* **DO NOT** disassemble the circuit
* **DO NOT** use any chemical or liquid cleaners when cleaning the robot
* If smoke or strange smells are detected, immediately stop using the robot and call us.

# INDEX

**A**

AVR Studio 6

**B**

Battery 4, 5, 10, 13, 14

**C**

Charger 13, 14

Charging 13

**I**

ISP Connector 4, 12

**M**

Maintenance 13

Memory 12

**O**

Overflow 12

**P**

Power On/Off 4, 5, 10

PonyProg 6

**S**

Software 2, 4, 6, 8, 11

Speed 7, 8

Switch 5, 10, 13

**T**

Troubleshooting 10, 12

**W**

Warning 14