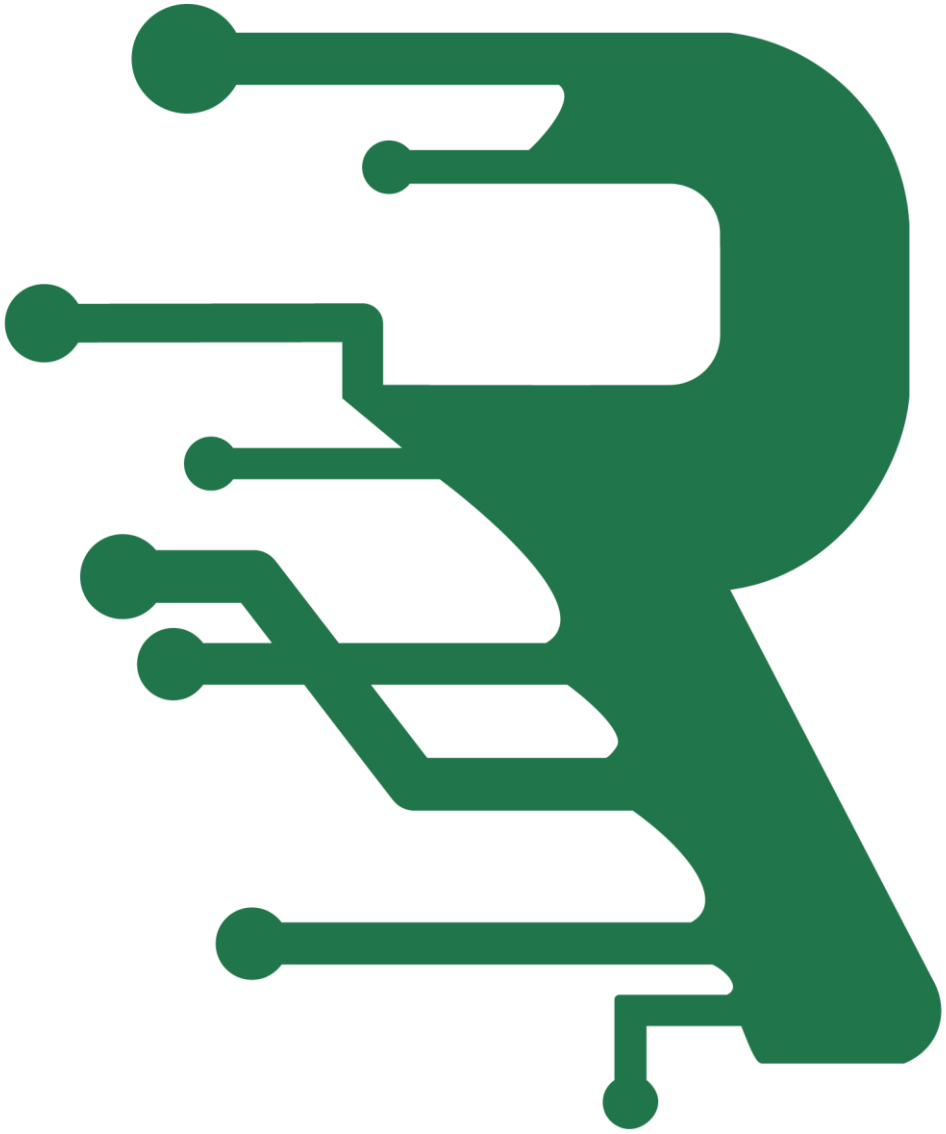

Roobopoli

MANUAL of calibrazione of ROOBOKART sensors

Summary

Revision history

Revision	Date	Description
Reva	18/05/2020	First release
RevB	24/06/2022	Roobokart v3 Support

Hardware configuration

Before moving on to Roobokart programming, you must verify that the following hardware specifications are met.

Front sensor board height

This adjustment procedure must be carried outonly once during the assembly of the Roobokart and before any other software adjustment.

The sensor board must be positioned on the front of the Roobokart using the appropriate spacers.

For Roobokart v1-v2 models the ideal distance is between 5 and 15 mm

For Roobokart v3 models the ideal distance is between 15 and 20 mm

The distance between the IR sensor and the road must be measured.

Changing the height of the sensor board can affect other adjustments.

Position of the battery pack

This adjustment procedure must be carried outonly once during the assembly of the Roobokart and before adjusting the PID.

The battery pack should be positioned as close as possible to the front wheels and as far away as possible from the ball bearing in contact with the road if present. The ideal place to place it on is above the axle of the front wheels.

If alternative battery packs are used, prefer energy sources that are as light as possible.

Use short and good quality power cables, a good indicator of the quality of the cable is the thickness that must be sufficient to ensure the correct power supply of the motors.

A more appropriate method for choosing the cable is to measure the maximum amount of current it is capable of carrying at 5 volts.

It is advisable to evaluate the use of the different connectors available on the main board for powering the motors according to your needs.

Firmware configuration

Once the Roobokart has been assembled, before compiling and programming the microcontroller with the obtained binary file, it is necessary to identify the correct values of some hardware-dependent parameters that may differ from vehicle to vehicle.

These parameters are described below and a method of evaluating the correct value is suggested for each of them.

Braking Speed

This adjustment procedure must be carried out if at the traffic light, the Roobokart does not stop in time or retreats.

Due to inertia, when the vehicle is moving, it is not enough to set the speed of the engines to zero to stop it quickly when it is required, as happens for example near the traffic light, but it is necessary to impose a speed opposite to the one you were holding, just enough to ensure that the driving force is able to quickly cancel the inertia without then inducing a change of the direction of travel.

This braking force is obtained by imposing a speed opposite to that of cruising. This speed value is engine-dependent and may differ from vehicle to vehicle, so it should be determined.

A typical value for Roobokart v1-v2 varies between -10 and -5 while for v3 it varies between -15 and -10. To find the correct value, proceed in this way. Load on the Roobokart a code that makes it travel a straight path for a few seconds and then apply a speed opposite to the previous one.

Threshold for traffic light colors

This adjustment procedure must be carried outif the preset default values do not correctly detect the colors of the traffic light.

To ensure that through the color sensor it is possible to detect the correct color of the horizontal traffic light and in time to be able to brake the vehicle, it is necessary to set a threshold value such that the color can be detected a few moments before the Roobokart is with the color sensor that looks at the RGB LED strip.

To measure the threshold to be set, place the stationary vehicle with the sensor about 1cm from the traffic light. Load the code that reads the red and green colors and print the values read on PC via USB.

Set read values by replacing them with default values in the file

roobopoli/roobokart/roobokart_def.h

```
#define COLOR_THRESHOLD 0.3f
```

It is possible that the reading value must be modified to ensure correct operation in all light situations and with all the different brands of LEDs and intensity of light emitted. A typical value varies between 0.3f and 0.5f.

It is advisable to feed the LED strips with a stabilized power supply to avoid abnormal fluctuations in brightness.

Motor speed in nav_mod

This adjustment procedure must be carried out if the Roobokart does not move or freezes during navigation

Adjust the speed of the motors before those of the PID control.

The adjustment of these parameters allows to manage the different types of motors on the market and to simplify the PID adjustment.

To find the minimum browsing speed, load the code that spins a single wheel at low speed, and increase it if the Roobokart does not stably

make more than one turn on itself. Once you find it, increment it by one or two units. A typical value is between 27 and 32 with low reduction engines. To find the cruising speed of navigation just set a speed greater than the minimum speed, typically 5 units are sufficient, you can increase it further if the navigation does not become unstable.

It is therefore necessary to set I and thresholds:

```
#define MIN_NAV_SPEED 30
#define CRUISE_NAV_SPEED 35
```

Which is located in the file roobopoli/roobokart/roobokart_def.h

PID control constants in nav_mod

This procedure must be performed after the calibration of the IR sensors, if the Roobokart does not correctly follow the white line on the right of the lane or when starting it rotates on itself.

The Roobokart at startup may begin to rotate on itself. Most likely this is due to an incorrect connection of the motors. In particular, the connections of the right and left motors could be reversed and/or the polarities reversed.

In the absence of a mathematical model, we proceed empirically to determine the values of the three constants.

Set Kp to its default value, the released one, and Ki= 0 Kd = 0.

If with these settings, the robot oscillates a lot and then loses the line decrease Kp by 1.

If, on the other hand, the robot moves away from the line to the right or left without oscillating, increase by 1

Repeat this operation, until, although oscillating, the robot manages to follow the line for a straight stretch and in a soft curve.

Once the Kp value is set, set Ki to 1 and increment by 1 until the robot can perform tighter turns. It is advisable to keep low values, possibly less than 10 and if necessary try to increase by 0.1. Decrease by 1 if the oscillation increases until the line is lost.

Also fixed the value of Ki, set kd to 0. 1, increment by 0. 1 until the oscillation subsides. Keep to low values, possibly less than 5, and if necessary increase by 0.1. Decrease by 0.1 if the robot fails to perform the tightest turns.

This procedure stems from the experience and observation of the behavior of the Roobokart. Other procedures may be more effective.

It is therefore necessary to set I and thresholds:

```
#define NAV_KP 50.0f
 #define NAV_KI 8.0f
#define NAV_KD 0.0f
```

Which is located in the file roobopoli/roobokart/roobokart_def.h

IR Sensor Calibration

This procedure should be performed once at first start or if a degradation of the analog sensors is suspected.

Please note that, as optical and analog sensors, IR sensors need calibration to make them efficient in all situations.
To perform the calibration, load the code that reads and prints the analog values directly from the IR sensors, and then update the parameters in the file: roobopoli/roobokart/roobokart_def.h

Observing the value read by the left IR sensor, place the aforementioned sensor above the white line, without tilting the Roobokart and change the parameter:

```
#Define IR_L_MIN_VALUE 0.04f
```

Observing the value read by the left IR sensor, place the aforementioned sensor above the black road, without tilting the Roobokart and change the parameter:

```
#define IR_L_MAX_VALUE 1.0f
```

Observing the value read by the right IR sensor, place the aforementioned sensor above the white line, without tilting the Roobokart and change the parameter:

```
#define IR_R_MIN_VALUE 0. 04f
```

Observing the value read by the right IR sensor, place the aforementioned sensor above the black road, without tilting the Roobokart and changing the parameter:

```
#define IR_R_MAX_VALUE 1.0f
```

Observing the value read by the central IR sensor, place the aforementioned sensor above the blue line of the sign, without tilting the Roobokart and changing the parameter:

```
#define IR_C_MIN_VALUE 0. 04f
```

Observing the value read by the right IR sensor, place the aforementioned sensor above the black road, without tilting the Roobokart and changing the parameter:

```
#define IR_C_MAX_VALUE 1.0f
```

The values read by the IR sensor are ideally between 0.0f and 1.0f, however the type of material and the distance of the surface interfere with these parameters, so it is necessary to perform the calibration that compensates for this discrepancy.

Right and left IR sensor threshold

This value is used to distinguish the white line from the black line.

Recall that the right sensor, in the basic mission is used for reading the line from the right, while for reading the value of the road sign both the right and the left are used.

The parameter to be modified is:

```
#define IR_THRESHOLD 0. 5f
```

Which is located in the file: roobopoli/roobokart/roobokart_def.h

Manual calibration of the right and left IR sensors makes the 0.5f value suitable for most situations.

It is advisable to keep the signs and the road surface clean.

Central IR sensor threshold

Recall that the central sensor, in the basic mission serves exclusively to recognize the presence of the road sign, through the blue band.

You must then set the parameter:

```
#define ROADSIGN_DETECTION_THRESHOLD_DEFAULT 0.5f
```

Which is located in the file roobopoli/roobokart/roobokart_def.h

Manual calibration of the right and left IR sensors makes the 0.5f value suitable for most situations.

It is advisable to keep road signs clean and to check that the blue color is light enough to be detected correctly.

Line Follow Point

The set point is the value that the PID controller tries to chase.

The automatic calibration system that takes place during post_mode makes the 0.5f value suitable for most situations.

It is therefore necessary to set the threshold to:

```
#define LINE_FOLLOWER_SP 0.5f
```

Which is located in the file roobopoli/roobokart/roobokart_def.h

You can vary this value between 0.4f and 0.6f to correct the direction of the Roobokart if it has a divergence to the right or left.

In the Roobokart v3 version the value may have to be set to 0.4f if you want to significantly reduce the height of the sensor board, this is because there are multiple emitters on the right sensor that at close distances can distort the measurement.

Adjusting the gyroscope KP

In case the Roobokart, after proper alignment with the road sign, does not maintain direction, increase the value by 0.1f as long as necessary.

In the event that the Roobokart, after a correct alignment with the road sign, begins to oscillate decrease the value of 0.1f as long as necessary.

A typical value is between 1.0f and 5.0f.

It is therefore necessary to set thethreshold to:

```
#define RSA_KP 1.2f
```

Which is located in the file roobopoli/roobokart/roobokart_def.h

Motor speed in rsa_modand

This adjustment procedure must be carried out if the Roobokart does not align correctly with the road sign

In case the Roobokart freezes, increase the speed of 1 until the alignment procedure finishes successfully.

In case the Roobokart begins to read the sign without completing the alignment, decrease the speed by 1 until the procedure is successful.

It is therefore necessary to set the threshold to:

```
#define CRUISE_RSA_SPEED 30
```

Which is located in the file roobopoli/roobokart/roobokart_def.h

In case of particular difficulties in the configuration of this parameter, it may be necessary to adjust the braking speed more carefully.

Motor speed in TRAFFICLIGHT_mode

This adjustment procedure must be carried out if the Roobokart does not stop in time at the traffic light or fails to restart after the green.

After adjusting the threshold of the color sensor and the braking speed appropriately.

In case the Roobokart does not stop in time at the traffic light during the red, decrease the tracking speed by 1 as long as necessary. A typical

value is between 30 and 35. If you have difficulty configuring this parameter, check the braking speed.

It is therefore necessary to set the threshold to:

```
#define SEEK_TL_SPEED 30
```

Which is located in the file roobopoli/roobokart/roobokart_def.h

In the event that the Roobokart, after stopping at red, if the green snap, does not start, increase the escape velocity by 1 as long as necessary. If you have difficulty configuring this parameter, check the color recognition threshold configuration.

It is therefore necessary to set the threshold to:

```
#define ESCAPE_TL_SPEED 40
```

Which is located in the file roobopoli/roobokart/roobokart_def.h