GLOSSARY

F1Tenth: Scaled down F1 racing car in 1:10 scale.

Autonomous Vehicle (AV): Self-driving vehicle that operates without human intervention, using sensors and AI for perception, decision-making, and control.

Espressif IoT Development Framework (ESP-IDF): Open-source software development framework for building applications on ESP32 and ESP8266 microcontrollers.

Free Real-Time Operating System (FreeRTOS): It is an open-source, lightweight OS for embedded systems with real-time requirements, enabling reliable and responsive applications.

Firmware: Embedded software in electronic devices that controls hardware functionality.

Robot Operating System 2 (ROS 2): Open-source framework for building robot applications, enabling communication and control of robotic systems.

Micro-ROS: Lightweight ROS implementation for microcontrollers, enabling distributed communication and control in small-scale robotic devices.

General Purpose Input Output (GPIO): Programmable pins on microcontrollers/computers for versatile connection and control of external devices.

UDP transport: Lightweight, connectionless protocol for fast communication, used in real-time applications where speed is prioritized over reliability.

Secure Shell (SSH): SSH is a cryptographic network protocol that provides a secure and encrypted means of connecting to remote servers or devices over an insecure network.

Light Detection and Ranging (LIDAR): LIDAR is a remote sensing technology that uses laser light to measure distances and create detailed 3D maps of the surrounding environment.

Mini-PC: A compact and small-sized personal computer that is designed to provide functionality similar to a traditional desktop computer while occupying less space.

ESP32: Low-power, dual-core microcontroller with Wi-Fi, Bluetooth, and extensive peripheral support, widely used for IoT applications.

Electronic Speed Controller (ESC): An Electronic Speed Controller is a device used to control the speed and direction of an electric motor, commonly found in applications such as drones, RC cars, and electric vehicles.

Battery Eliminator Circuit (BEC): An electronic circuit commonly used in RC (radio control) applications to provide regulated power to the receiver and other low-power components without the need for a separate receiver battery.

Wall Following: A navigation technique commonly used in robotics and autonomous systems where a robot or vehicle maintains a certain distance or parallel alignment with a nearby wall or obstacle while moving along a path.

Follow the Gap Method (FGM): The Follow the Gap Method is a navigation for obstacle avoidance used in robotics and autonomous systems to drive through environments by following open spaces or gaps between obstacles.

LIDAR Filter: Algorithm or technique used to refine raw LIDAR data by removing noise and outliers, enhancing accuracy and reliability for applications such as object detection and mapping.

REFERENCES

Building the F1TENTH Car (2022) f1tenth.readthedocs.io. Available at: https://f1tenth.readthedocs.io/en/foxy_test/getting_started/build_car/index.html#building-the-f1tenth-car.

Raistiawan, A. T. (2022). "Navigation Algorithm for Automatic Robot Vehicle Based on Robot Operating System 2." Swiss German University.

Putera, V. (2018) FURTHER DEVELOPMENT OF AN AUTONOMOUS MOBILE ROBOT NAVIGATION SYSTEM USING ROBOT OPERATING SYSTEM. Swiss German University.

Bill of Materials (2022) f1tenth.readthedocs.io. Available at: https://f1tenth.readthedocs.io/en/foxy_test/getting_started/build_car/bom.html.

S. Macenski, T. Foote, B. Gerkey, C. Lalancette, W. Woodall, “Robot Operating System 2: Design, architecture, and uses in the wild,” Science Robotics vol. 7, May 2022.

Babu, V. S. and Behl, M. (2019). "ROS F1/10 Autonomous Racing Simulator." In: ROSCon 2019, Computer Science, Link Lab, University of Virginia. Available at: https://roscon.ros.org/2019/talks/roscon2019_f110th.pdf.

Understanding ROS 2 nodes (2022) docs.ros.org. Available at: https://docs.ros.org/en/foxy/Tutorials/Understanding-ROS2-Nodes.html.

Understanding ROS 2 topics (2022) docs.ros.org. Available at: https://docs.ros.org/en/foxy/Tutorials/Topics/Understanding-ROS2-Topics.html.

Understanding ROS 2 services (2022) docs.ros.org. Available at: https://docs.ros.org/en/foxy/Tutorials/Services/Understanding-ROS2-Services.html.

Understanding ROS 2 parameters (2022) docs.ros.org. Available at: https://docs.ros.org/en/foxy/Tutorials/Parameters/Understanding-ROS2Parameters.html.

Rothmeyer, A. (2021) All About LiDAR and How it Helps During a Robotics Boom, automation.com. Available at: https://www.automation.com/en-us/articles/may2021/lidar-helps-during-robotics-boom (Accessed: 21 April 2022).

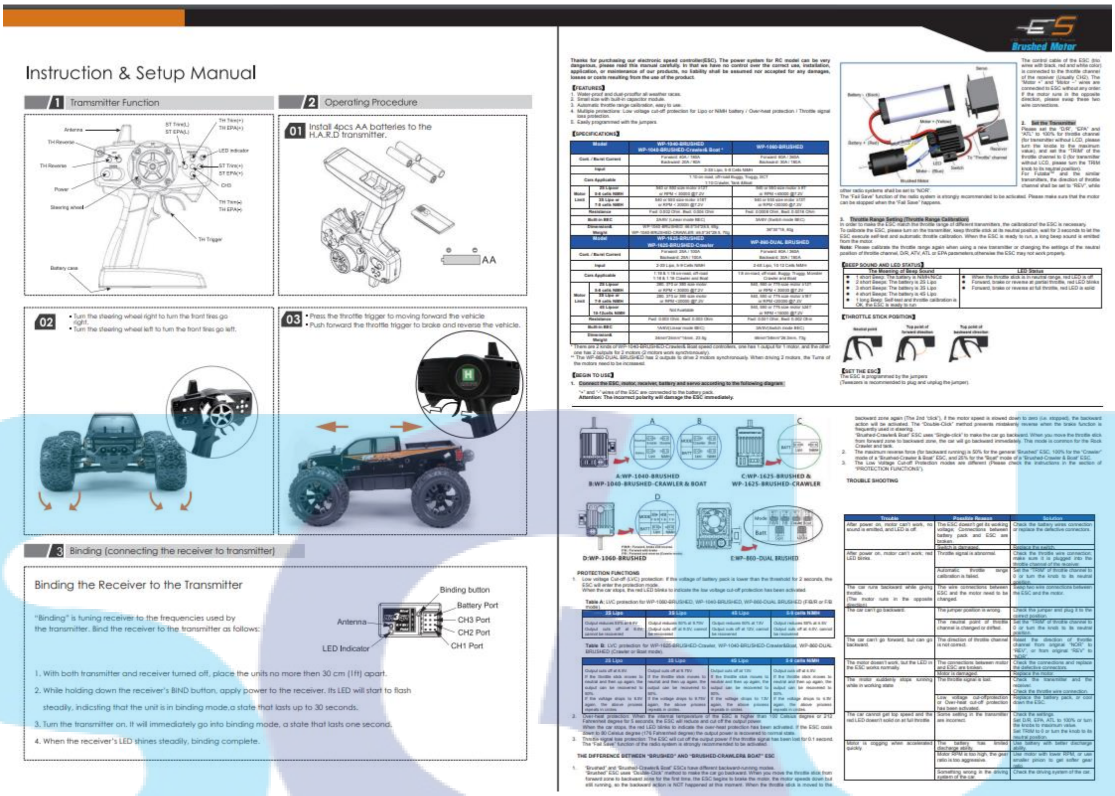
‘Lab 3: Wall Following Learning Outcomes Review of PID in the time domain Wall Following’ (no date), pp. 3–6.

Sears-Collins, A. (2020) The Bug2 Algorithm for Robot Motion Planning, automaticaddison.com. Available at: https://automaticaddison.com/the-bug2-algorithm-for-robot-motion-planning/ (Accessed: 20 April 2022).

Demir, M. and Sezer, V. (2017). "Improved Follow the Gap Method for obstacle avoidance." In: 2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM), Munich, Germany, pp. 1435-1440. doi: 10.1109/AIM.2017.8014220.

Bosello, M., Tse, R., & Pau, G. (2022). "Train in Austria, Race in Montecarlo: Generalized RL for Cross-Track F1 tenth LIDAR-Based Races." In: 2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC). IEEE.

APPENDIX A – DATASHEET



Instruction & Setup Manual

1 Tramsmitter Function

Antenna
TH Reverse
TH Reverse
Power
Steering wheel
Battery case
ST Trim(L)
ST EPA(L)
TH Trim(+)
TH EPA(+)
LED indicator
ST Trim(+)
ST EPA(+)
CH3
TH Trim(-)
TH EPA(-)
TH Trigger

2 Operating Procedure

01 Install 4pcs AA batteries to the H.A.R.D transmitter.

AA

02
- Turn the steering wheel right to turn the front tires go right.
- Turn the steering wheel left to turn the front tires go left.

03
- Press the throttle trigger to moving forward the vehicle
- Push forward the throttle trigger to brake and reverse the vehicle.

3 Binding (connecting the receiver to transmitter)

Binding the Receiver to the Transmitter

"Binding" is tuning receiver to the frequencies used by the transmitter. Bind the receiver to the transmitter as follows:

1. With both transmitter and receiver turned off, place the units no more then 30 cm (1ft) apart.
2. While holding down the receiver's BIND button, apply power to the receiver. Its LED will start to flash steadily, indicsting that the unit is in binding mode,a state that lasts up to 30 seconds.
3. Turn the transmitter on. It will immediately go into binding mode, a state that lasts one second.
4. When the receiver's LED shines steadily, binding complete.

Thanks for purchasing our electronic speed controller(ESC). The power system for RC model can be very dangerous, please read this manual carefully. In that we have no control over the correct use, installation, application, or maintenance of our products, no liability shall be assumed nor accepted for any damages, losses or costs resulting from the use of the product.

【FEATURES】
1. Water-proof and dust-proof for all weather races.
2. Small size with built-in capacitor module.
3. Automatic throttle range calibration, easy to use.
4. Multiple protections: Low voltage cut-off protection for Lipo or NiMH battery / Over-heat protection / Throttle signal loss protection.
5. Easily programmed with the jumpers.

【SPECIFICATIONS】

Model		WP-1040-BRUSHED WP-1040-BRUSHED-Crawler& Boat *	WP-1060-BRUSHED
Cont. / Burst Current		Forward: 40A / 180A Backward: 20A / 90A	Forward: 60A / 360A Backward: 30A / 180A
Input		2-3S Lipo, 5-9 Cells NiMH	
Cars Applicable		1:10 on-road, off-road Buggy, Truggy, SCT 1:10 Crawler, Tank &Boat	
Motor Limit	2S Lipo or 5-6 cells NiMH	540 or 550 size motor ≥12T or RPM < 30000 @7.2V	540 or 550 size motor ≥8T or RPM <40000 @7.2V
	3S Lipo or 7-9 cells NiMH	540 or 550 size motor ≥18T or RPM < 20000 @7.2V	540 or 550 size motor ≥12T or RPM <30000 @7.2V
Resistance		Fwd: 0.002 Ohm, Bwd: 0.004 Ohm	Fwd: 0.0008 Ohm, Bwd: 0.0016 Ohm
Built-in BEC		2A/6V (Linear mode BEC)	3A/6V (Switch-mode BEC)
Dimension& Weight		WP-1040-BRUSHED: 46.5*34*28.5, 48g WP-1040-BRUSHED-CRAWLER: 46.5*34*28.5, 70g	36*30*18, 40g
Model		WP-1625-BRUSHED WP-1625-BRUSHED-Crawler	WP-860-DUAL BRUSHED
Cont. / Burst Current		Forward: 25A / 100A Backward: 25A / 100A	Forward: 60A / 360A Backward: 30A / 180A
Input		2-3S Lipo, 5-9 Cells NiMH	2-4S Lipo, 10-12 Cells NiMH
Cars Applicable		1:18 & 1:16 on-road, off-road 1:18 & 1:16 Crawler and Boat	1:8 on-road, off-road, Buggy, Truggy, Monster Crawler and Boat
Motor Limit	2S Lipo or 5-6 cells NiMH	280, 370 or 380 size motor or RPM < 30000 @7.2V	540, 550 or 775 size motor ≥12T or RPM < 30000 @7.2V
	3S Lipo or 7-9 cells NiMH	280, 370 or 380 size motor or RPM <20000 @7.2V	540, 550 or 775 size motor ≥18T or RPM <20000 @7.2V
	4S Lipo or 10-12cells NiMH	Not Available	540, 550 or 775 size motor ≥24T or RPM <15000 @7.2V
Resistance		Fwd: 0.003 Ohm, Bwd: 0.003 Ohm	Fwd: 0.001 Ohm, Bwd: 0.002 Ohm
Built-in BEC		1A/6V(Linear mode BEC)	3A/6V(Switch mode BEC)
Dimension& Weight		34mm*24mm*14mm, 23.8g	48mm*38mm*28.3mm, 73g

* There are 2 kinds of WP-1040-BRUSHED-Crawler& Boat speed controllers, one has 1 output for 1 motor, and the other one has 2 outputs for 2 motors (2 motors work synchronously).
** The WP-860-DUAL BRUSHED has 2 outputs to drive 2 motors synchronously. When driving 2 motors, the Turns of the motors need to be increased.

【BEGIN TO USE】
1. **Connect the ESC, motor, receiver, battery and servo according to the following diagram:**

"+" and "-" wires of the ESC are connected to the battery pack.
Attention: The incorrect polarity will damage the ESC immediately.

The control cable of the ESC (trio wires with black, red and white color) is connected to the throttle channel of the receiver (Usually CH2). The "Motor +" and "Motor –" wires are connected to ESC without any order. If the motor runs in the opposite direction, please swap these two wire connections.

2. **Set the Transmitter**
Please set the "D/R", "EPA" and "ATL" to 100% for throttle channel (for transmitter without LCD, please turn the knobs to the maximum value), and set the "TRIM" of the throttle channel to 0 (for transmitter without LCD, please turn the TRIM knob to its neutral position).
For Futaba™ and the similar transmitters, the direction of throttle channel shall be set to "REV", while other radio systems shall be set to "NOR".
The "Fail Save" function of the radio system is strongly recommended to be activated. Please make sure that the motor can be stopped when the "Fail Save" happens.

3. **Throttle Range Setting (Throttle Range Calibration)**
In order to make the ESC match the throttle range of different transmitters, the calibrationof the ESC is necessary.
To calibrate the ESC, please turn on the transmitter, keep throttle stick at its neutral position, wait for 3 seconds to let the ESC execute self-test and automatic throttle calibration. When the ESC is ready to run, a long beep sound is emitted from the motor.
Note: Please calibrate the throttle range again when using a new transmitter or changing the settings of the neutral position of throttle channel, D/R, ATV, ATL or EPA parameters,otherwise the ESC may not work properly.

【BEEP SOUND AND LED STATUS】

The Meaning of Beep Sound	LED Status
• 1 short Beep: The battery is NiMH/NiCd • 2 short Beeps: The battery is 2S Lipo • 3 short Beeps: The battery is 3S Lipo • 4 short Beeps: The battery is 4S Lipo • 1 long Beep: Self-test and throttle calibration is OK, the ESC is ready to run	• When the throttle stick is in neutral range, red LED is off • Forward, brake or reverse at partial throttle, red LED blinks • Forward, brake or reverse at full throttle, red LED is solid

【THROTTLE STICK POSITION】

Neutral point
Top point of forward direction
Top point of backward direction

【SET THE ESC】
The ESC is programmed by the jumpers
(Tweezers is recommended to plug and unplug the jumper).

A:WP-1040-BRUSHED
B:WP-1040-BRUSHED-CRAWLER & BOAT

C:WP-1625-BRUSHED &
WP-1625-BRUSHED-CRAWLER

D:WP-1060-BRUSHED

E:WP-860-DUAL BRUSHED

PROTECTION FUNCTIONS
1. Low voltage Cut-off (LVC) protection: If the voltage of battery pack is lower than the threshold for 2 seconds, the ESC will enter the protection mode.
When the car stops, the red LED blinks to indicate the low voltage cut-off protection has been activated.

Table A: LVC protection for WP-1060-BRUSHED, WP-1040-BRUSHED, WP-860-DUAL BRUSHED (F/B/R or F/B mode).

2S Lipo	3S Lipo	4S Lipo	5-9 cells NiMH
Output reduces 50% at 6.5V Output cuts off at 6.0V, cannot be recovered	Output reduces 50% at 9.75V Output cuts off at 9.0V, cannot be recovered	Output reduces 50% at 13V Output cuts off at 12V, cannot be recovered	Output reduces 50% at 4.5V Output cuts off at 4.0V, cannot be recovered

Table B: LVC protection for WP-1625-BRUSHED-Crawler, WP-1040-BRUSHED-Crawler&Boat, WP-860-DUAL BRUSHED (Crawler or Boat mode).

2S Lipo	3S Lipo	4S Lipo	5-9 cells NiMH
Output cuts off at 6.5V. If the throttle stick moves to neutral and then up again, the output can be recovered to 50%. If the voltage drops to 6.5V again, the above process repeats in circles.	Output cuts off at 9.75V If the throttle stick moves to neutral and then up again, the output can be recovered to 50%. If the voltage drops to 9.75V again, the above process repeats in circles.	Output cuts off at 13V If the throttle stick moves to neutral and then up again, the output can be recovered to 50%. If the voltage drops to 13V again, the above process repeats in circles.	Output cuts off at 4.5V If the throttle stick moves to neutral and then up again, the output can be recovered to 50%. If the voltage drops to 4.5V again, the above process repeats in circles.

2. Over-heat protection: When the internal temperature of the ESC is higher than 100 Celsius degree or 212 Fahrenheit degree for 5 seconds, the ESC will reduce and cut off the output power.
When the car stops, the red LED blinks to indicate the over-heat protection has been activated. If the ESC cools down to 80 Celsius degree (176 Fahrenheit degree) the output power is recovered to normal state.
3. Throttle signal loss protection: The ESC will cut off the output power if the throttle signal has been lost for 0.1 second. The "Fail Save" function of the radio system is strongly recommended to be activated.

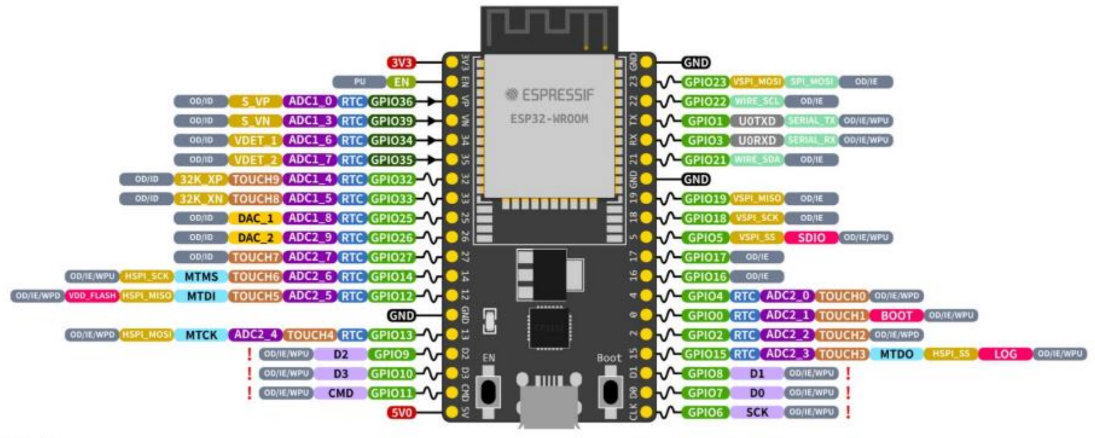
THE DIFFERENCE BETWEEN "BRUSHED" AND "BRUSHED-CRAWLER& BOAT" ESC

1. "Brushed" and "Brushed-Crawler& Boat" ESCs have different backward-running modes.
"Brushed" ESC uses "Double-Click" method to make the car go backward. When you move the throttle stick from forward zone to backward zone for the first time, the ESC begins to brake the motor, the motor speeds down but still running, so the backward action is NOT happened at this moment. When the throttle stick is moved to the backward zone again (The 2nd "click"), if the motor speed is slowed down to zero (i.e. stopped), the backward action will be activated. The "Double-Click" method prevents mistakenly reverse when the brake function is frequently used in steering.
"Brushed-Crawler& Boat" ESC uses "Single-click" to make the car go backward. When you move the throttle stick from forward zone to backward zone, the car will go backward immediately. This mode is common for the Rock Crawler and tank.
2. The maximum reverse force (for backward running) is 50% for the general "Brushed" ESC, 100% for the "Crawler" mode of a "Brushed-Crawler & Boat" ESC, and 25% for the "Boat" mode of a "Brushed-Crawler & Boat" ESC.
3. The Low Voltage Cut-off Protection modes are different (Please check the instructions in the section of "PROTECTION FUNCTIONS").

TROUBLE SHOOTING

Trouble	Possible Reason	Solution
After power on, motor can't work, no sound is emitted, and LED is off.	The ESC doesn't get its working voltage; Connections between battery pack and ESC are broken.	Check the battery wires connection or replace the defective connectors.
	Switch is damaged.	Replace the switch.
After power on, motor can't work, red LED blinks.	Throttle signal is abnormal.	Check the throttle wire connection, make sure it is plugged into the throttle channel of the receiver.
	Automatic throttle range calibration is failed.	Set the "TRIM" of throttle channel to 0 or turn the knob to its neutral position.
The car runs backward while giving throttle. (The motor runs in the opposite direction)	The wire connections between ESC and the motor need to be changed.	Swap two wire connections between the ESC and the motor.
The car can't go backward.	The jumper position is wrong.	Check the jumper and plug it to the correct position.
	The neutral point of throttle channel is changed or drifted.	Set the "TRIM" of throttle channel to 0 or turn the knob to its neutral position.
The car can't go forward, but can go backward.	The direction of throttle channel is not correct.	Reset the direction of throttle channel from original "NOR" to "REV", or from original "REV" to "NOR".
The motor doesn't work, but the LED in the ESC works normally.	The connections between motor and ESC are broken.	Check the connections and replace the defective connectors.
	Motor is damaged.	Replace the motor.
The motor suddenly stops running while in working state.	The throttle signal is lost.	Check the transmitter and the receiver. Check the throttle wire connection.
	Low voltage cut-off protection or Over-heat cut-off protection has been activated.	Replace the battery pack, or cool down the ESC.
The car cannot get top speed and the red LED doesn't solid on at full throttle	Some setting in the transmitter are incorrect.	Check the settings. Set D/R, EPA, ATL to 100% or turn the knobs to maximum value. Set TRIM to 0 or turn the knob to its neutral position.
Motor is cogging when accelerated quickly.	The battery has limited discharge ability.	Use battery with better discharge ability.
	Motor RPM is too high, the gear ratio is too aggressive.	Use motor with lower RPM, or use smaller pinion to get softer gear ratio.
	Something wrong in the driving system of the car.	Check the driving system of the car.

ESP32-DevKitC



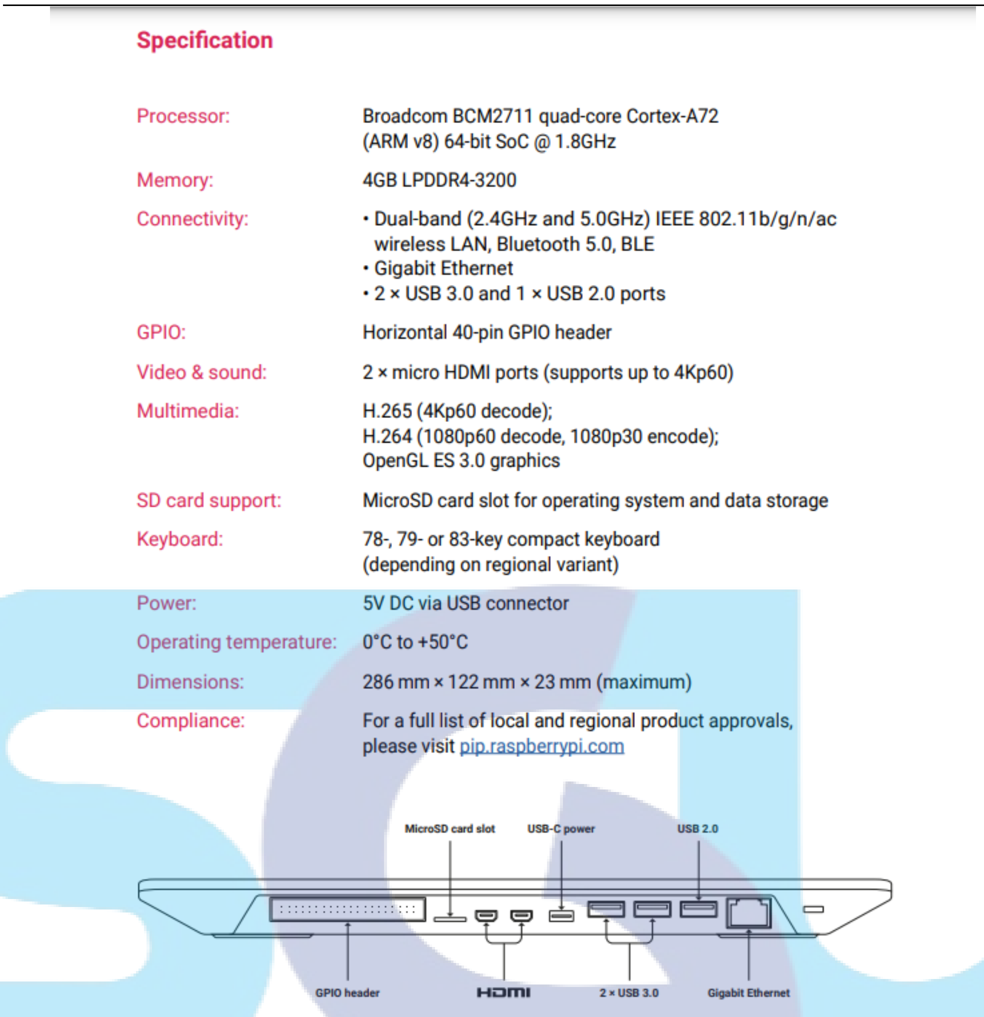
ESPRESSIF
ESP32-WROOM
EN
Boot
3V3 | 3V3
PU EN | EN
OD/ID S_VP ADC1_0 RTC GPIO36 | VP
OD/ID S_VN ADC1_3 RTC GPIO39 | VN
OD/ID VDET_1 ADC1_6 RTC GPIO34 | 34
OD/ID VDET_2 ADC1_7 RTC GPIO35 | 35
OD/ID 32K_XP TOUCH9 ADC1_4 RTC GPIO32 | 32
OD/ID 32K_XN TOUCH8 ADC1_5 RTC GPIO33 | 33
OD/ID DAC_1 ADC1_8 RTC GPIO25 | 25
OD/ID DAC_2 ADC2_9 RTC GPIO26 | 26
OD/ID TOUCH7 ADC2_7 RTC GPIO27 | 27
OD/IE/WPU HSPI_SCK MTMS TOUCH6 ADC2_6 RTC GPIO14 | 14
OD/IE/WPD VDD_FLASH HSPI_MISO MTDI TOUCH5 ADC2_5 RTC GPIO12 | 12
GND | GND
OD/IE/WPD HSPI_MOSI MTCK ADC2_4 TOUCH4 RTC GPIO13 | 13
OD/IE/WPU D2 GPIO9 | D2
! OD/IE/WPU D3 GPIO10 | D3
! OD/IE/WPU CMD GPIO11 | CMD
5V0 | 5V
GND | GND
23 | GPIO23 VSPI_MOSI SPI_MOSI OD/IE
22 | GPIO22 WIRE_SCL OD/IE
TX | GPIO1 U0TXD SERIAL_TX OD/IE/WPU
RX | GPIO3 U0RXD SERIAL_RX OD/IE/WPU
21 | GPIO21 WIRE_SDA OD/IE
GND | GND
19 | GPIO19 VSPI_MISO OD/IE
18 | GPIO18 VSPI_SCK OD/IE
5 | GPIO5 VSPI_SS SDIO OD/IE/WPU
17 | GPIO17 OD/IE
16 | GPIO16 OD/IE
4 | GPIO4 RTC ADC2_0 TOUCH0 OD/IE/WPD
0 | GPIO0 RTC ADC2_1 TOUCH1 BOOT OD/IE/WPU
2 | GPIO2 RTC ADC2_2 TOUCH2 OD/IE/WPD
15 | GPIO15 RTC ADC2_3 TOUCH3 MTDO HSPI_SS LOG OD/IE/WPU
D1 | GPIO8 D1 OD/IE/WPU !
D0 | GPIO7 D0 OD/IE/WPU !
CLK | GPIO6 SCK OD/IE/WPU !

ESP32 Specs
32-bit Xtensa® dual-core @240MHz
Wi-Fi IEEE 802.11 b/g/n 2.4GHz
BLuetooth 4.2 BR/EDR and BLE
520 KB SRAM (16 KB for cache)
448 KB ROM
34 GPIOs, 4x SPI, 3x UART, 2x I2C,
2x I2S, RMT, LED PWM, 1 host SD/eMMC/SDIO,
1 slave SDIO/SPI, TWAI®, 12-bit ADC, Ethernet

∿	PWM Capable Pin
→ GPIOX	GPIO Input Only
GPIOX	GPIO Input and Output
DAC_X	Digital-to-Analog Converter
DEBUG	JTAG for Debugging
FLASH	External Flash Memory (SPI)
ADCX_CH	Analog-to-Digital Converter
TOUCHX	Touch Sensor Input Channel
OTHER	Other Related Functions
SERIAL	Serial for Debug/Programming
ARDUINO	Arduino Related Functions
STRAP	Strapping Pin Functions

RTC	RTC Power Domain (VDD3P3_RTC)
GND	Ground
PWD	Power Rails (3V3 and 5V)
!	Pin Shared with the Flash Memory Can't be used as regular GPIO

GPIO STATE
WPU: Weak Pull-up (Internal)
WPD: Weak Pull-down (Internal)
PU: Pull-up (External)
IE: Input Enable (After Reset)
ID: Input Disabled (After Reset)
OE: Output Enable (After Reset)
OD: Output Disabled (After Reset)

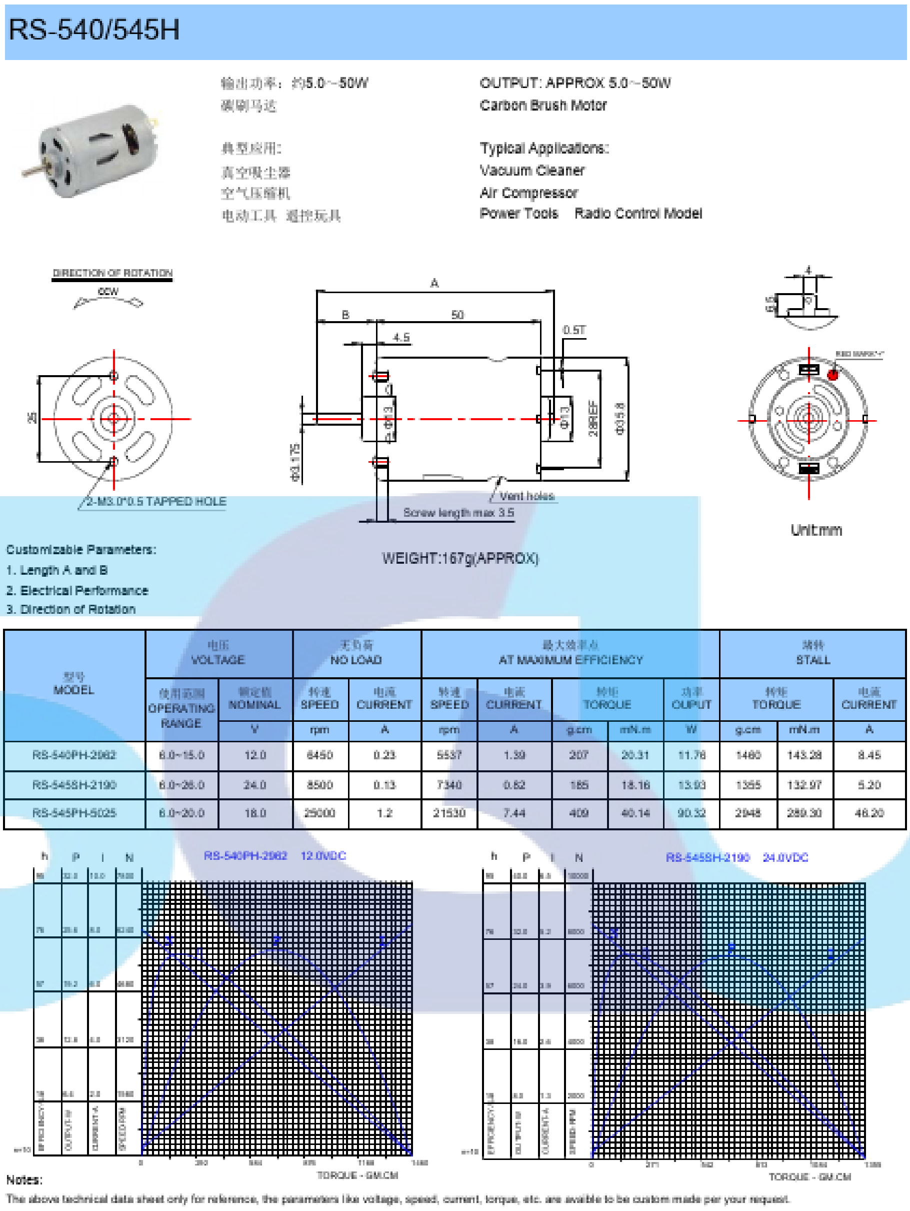


Specification

Processor:	Broadcom BCM2711 quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.8GHz
Memory:	4GB LPDDR4-3200
Connectivity:	• Dual-band (2.4GHz and 5.0GHz) IEEE 802.11b/g/n/ac wireless LAN, Bluetooth 5.0, BLE • Gigabit Ethernet • 2 × USB 3.0 and 1 × USB 2.0 ports
GPIO:	Horizontal 40-pin GPIO header
Video & sound:	2 × micro HDMI ports (supports up to 4Kp60)

Multimedia:	H.265 (4Kp60 decode);
	H.264 (1080p60 decode, 1080p30 encode);
	OpenGL ES 3.0 graphics

SD card support:	MicroSD card slot for operating system and data storage
Keyboard:	78-, 79- or 83-key compact keyboard (depending on regional variant)
Power:	5V DC via USB connector
Operating temperature:	0°C to +50°C
Dimensions:	286 mm × 122 mm × 23 mm (maximum)
Compliance:	For a full list of local and regional product approvals, please visit pip.raspberrypi.com



RS-540/545H

输出功率：约5.0~50W
碳刷马达

典型应用:
真空吸尘器
空气压缩机
电动工具 遥控玩具

OUTPUT: APPROX 5.0~50W
Carbon Brush Motor

Typical Applications:
Vacuum Cleaner
Air Compressor
Power Tools Radio Control Model

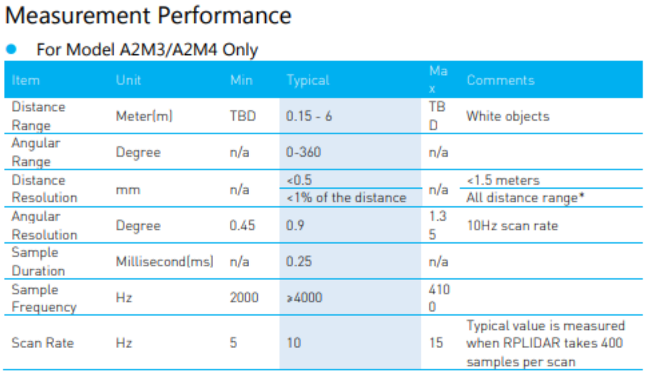
DIRECTION OF ROTATION
CCW
25
2-M3.0*0.5 TAPPED HOLE
A
B
50
4.5
0.5T
Φ13
Φ13
28REF
Φ35.8
Φ3.175
Vent holes
Screw length max 3.5
4
6.5
RED MARK"+"
Unit:mm

WEIGHT:167g(APPROX)

Customizable Parameters:
1. Length A and B
2. Electrical Performance
3. Direction of Rotation

型号 MODEL	电压 VOLTAGE		无负荷 NO LOAD		最大效率点 AT MAXIMUM EFFICIENCY					堵转 STALL		
	使用范围 OPERATING RANGE	额定值 NOMINAL	转速 SPEED	电流 CURRENT	转速 SPEED	电流 CURRENT	转矩 TORQUE		功率 OUPUT	转矩 TORQUE		电流 CURRENT
		V	rpm	A	rpm	A	g.cm	mN.m	W	g.cm	mN.m	A
RS-540PH-2962	6.0~15.0	12.0	6450	0.23	5537	1.39	207	20.31	11.76	1460	143.28	8.45
RS-545SH-2190	6.0~26.0	24.0	8500	0.13	7340	0.82	185	18.16	13.93	1355	132.97	5.20
RS-545PH-5025	6.0~20.0	18.0	25000	1.2	21530	7.44	409	40.14	90.32	2948	289.30	46.20

Notes:
The above technical data sheet only for reference, the parameters like voltage, speed, current, torque, etc. are avaible to be custom made per your request.



Measurement Performance

- For Model A2M3/A2M4 Only

Item	Unit	Min	Typical	Max	Comments
Distance Range	Meter(m)	TBD	0.15 - 6	TBD	White objects
Angular Range	Degree	n/a	0-360	n/a	
Distance Resolution	mm	n/a	<0.5 <1% of the distance	n/a	<1.5 meters All distance range*
Angular Resolution	Degree	0.45	0.9	1.35	10Hz scan rate
Sample Duration	Millisecond(ms)	n/a	0.25	n/a	
Sample Frequency	Hz	2000	≥4000	4100	
Scan Rate	Hz	5	10	15	Typical value is measured when RPLIDAR takes 400 samples per scan

Figure 2-1 RPLIDAR Performance

Note: the triangulation range system resolution changes along with distance, and the theoretical resolution change of RPLIDAR is shown as below:

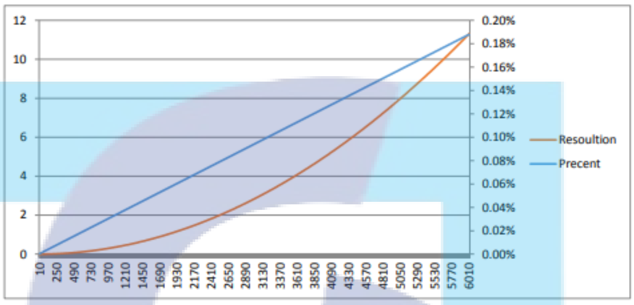
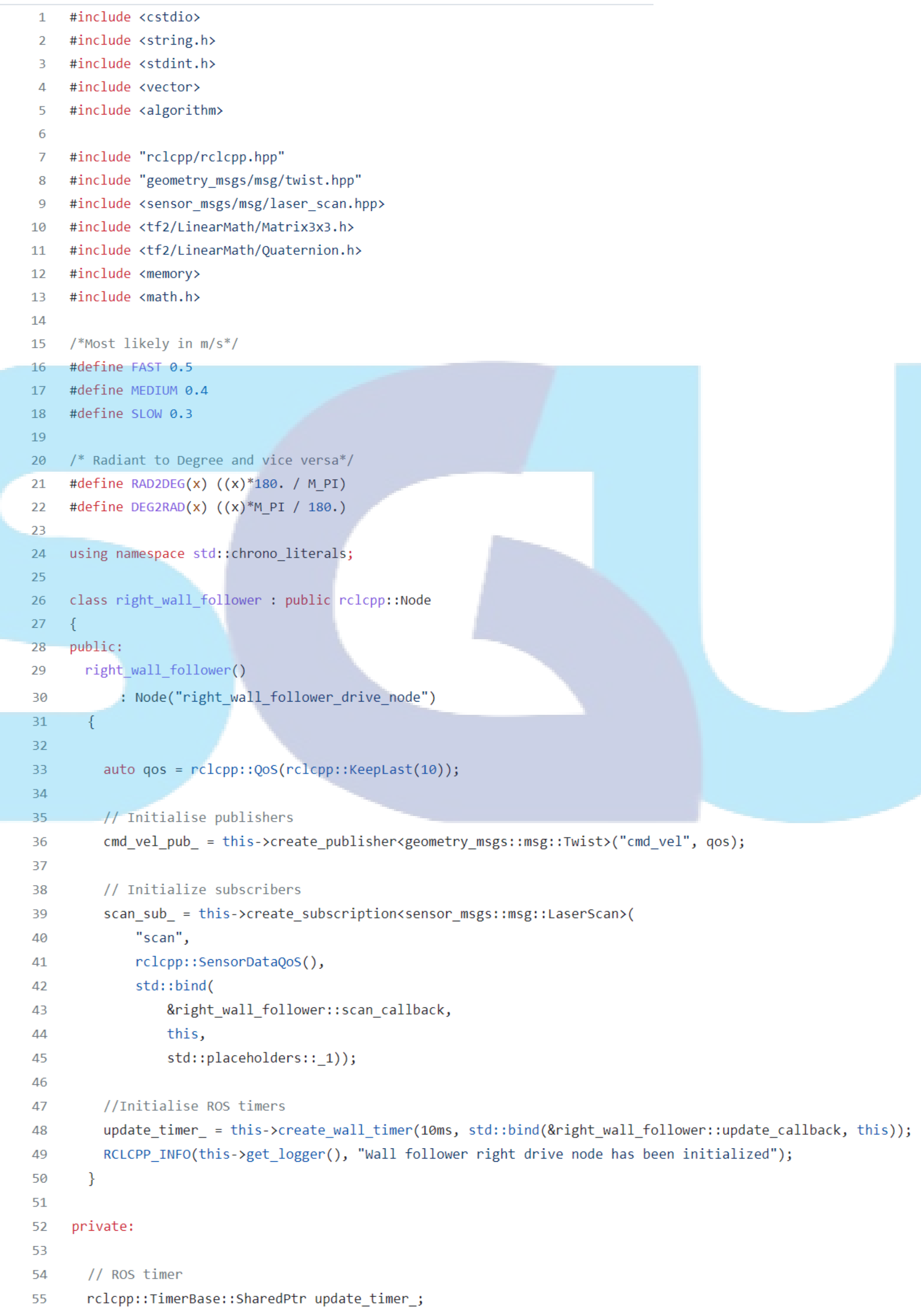
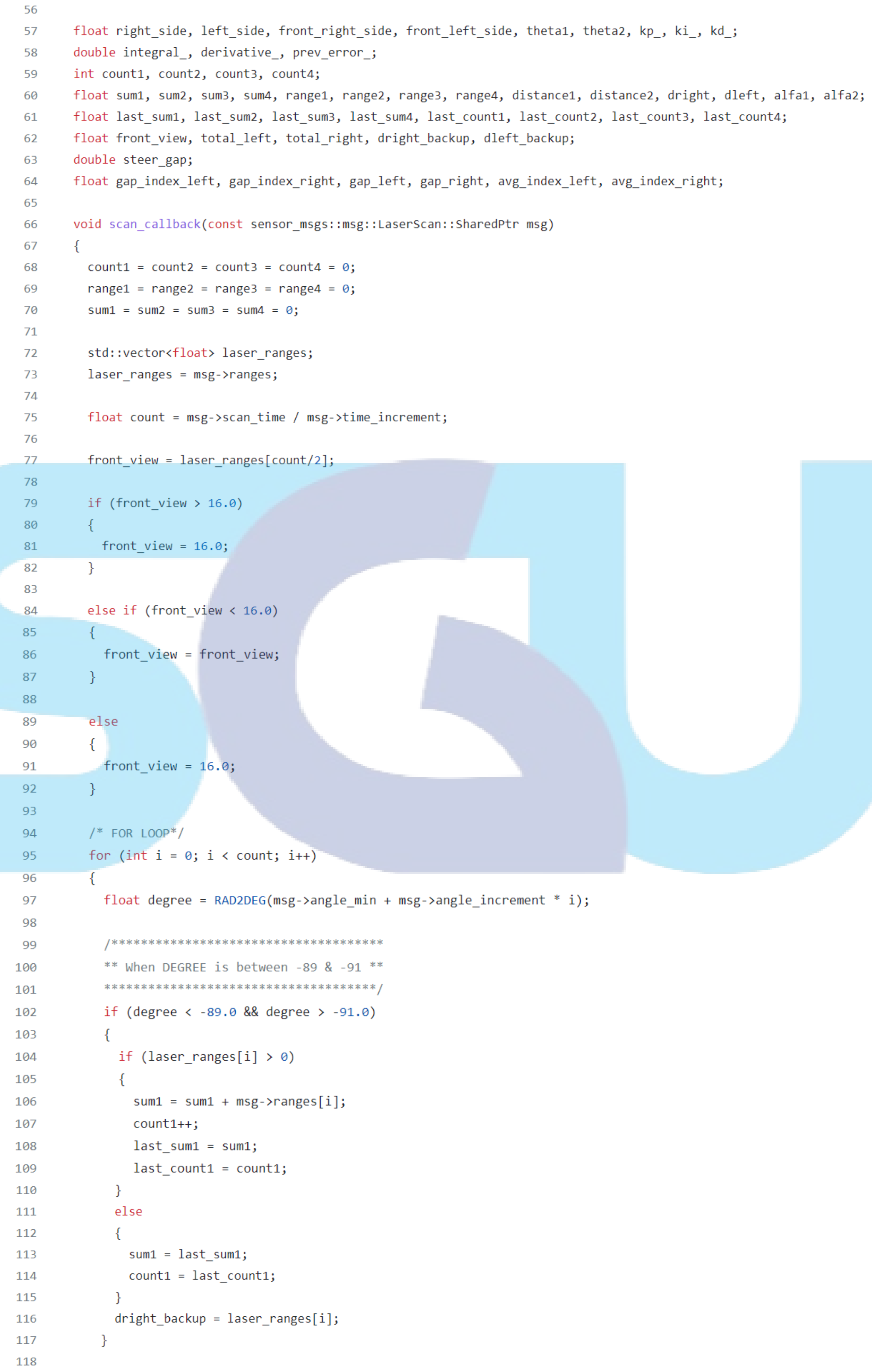


Figure 2-2 The Trend Graph of RPLIDAR Resolution

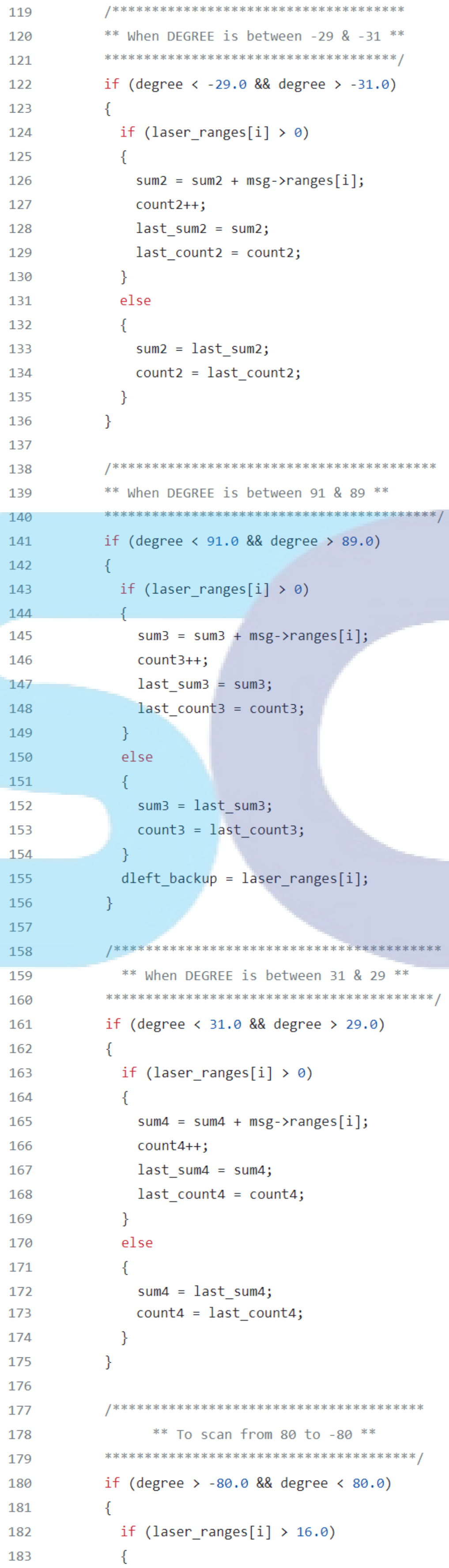
APPENDIX B – PROGRAM CODE



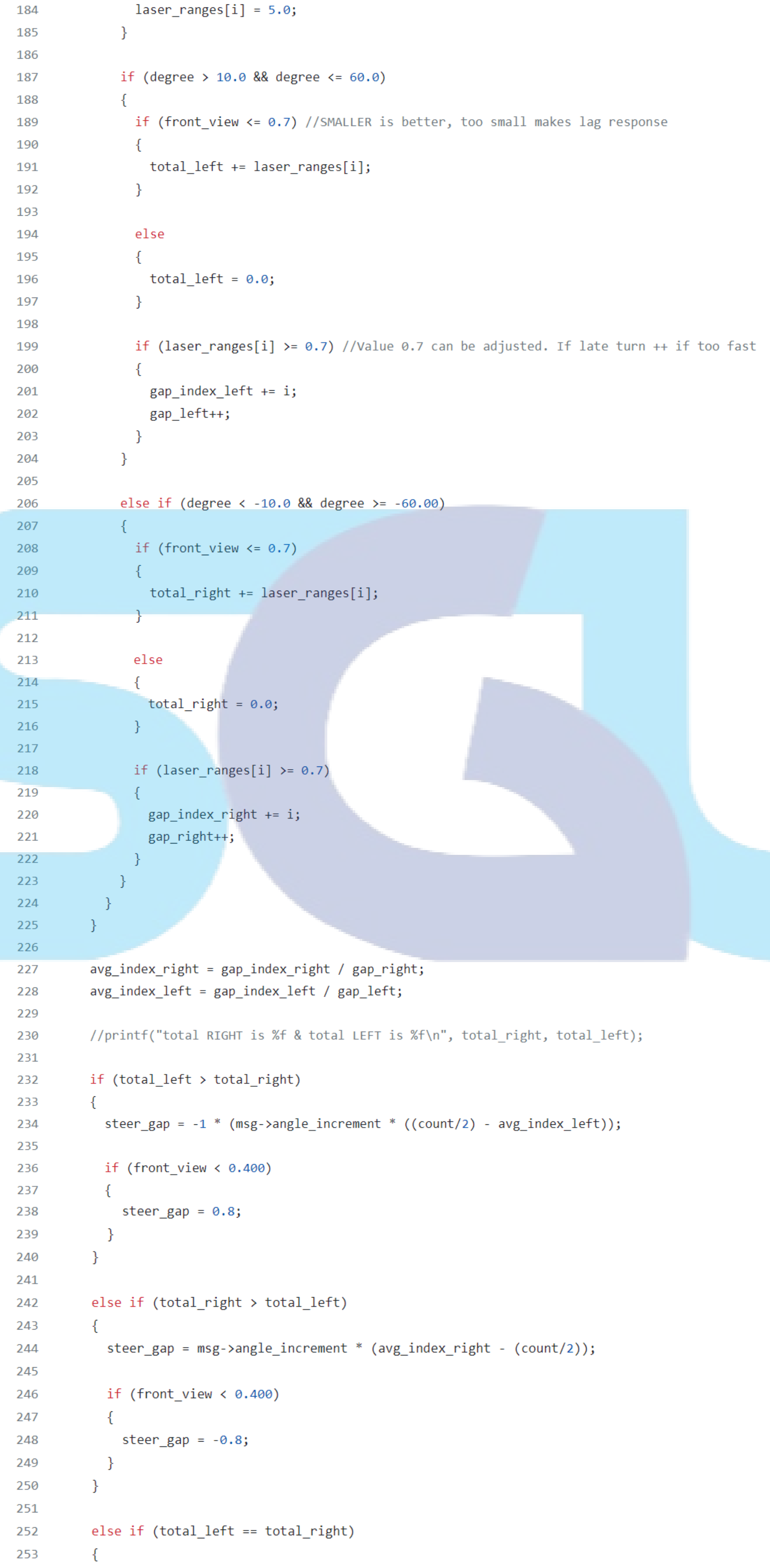
```
#include <cstdio>
#include <string.h>
#include <stdint.h>
#include <vector>
#include <algorithm>

#include "rclcpp/rclcpp.hpp"
#include "geometry_msgs/msg/twist.hpp"
#include <sensor_msgs/msg/laser_scan.hpp>
#include <tf2/LinearMath/Matrix3x3.h>
#include <tf2/LinearMath/Quaternion.h>
#include <memory>
#include <math.h>

/*Most likely in m/s*/
#define FAST 0.5
#define MEDIUM 0.4
#define SLOW 0.3

/* Radiant to Degree and vice versa*/
#define RAD2DEG(x) ((x)*180. / M_PI)
#define DEG2RAD(x) ((x)*M_PI / 180.)

using namespace std::chrono_literals;

class right_wall_follower : public rclcpp::Node
{
public:
  right_wall_follower()
      : Node("right_wall_follower_drive_node")
  {

    auto qos = rclcpp::QoS(rclcpp::KeepLast(10));

    // Initialise publishers
    cmd_vel_pub_ = this->create_publisher<geometry_msgs::msg::Twist>("cmd_vel", qos);

    // Initialize subscribers
    scan_sub_ = this->create_subscription<sensor_msgs::msg::LaserScan>(
        "scan",
        rclcpp::SensorDataQoS(),
        std::bind(
            &right_wall_follower::scan_callback,
            this,
            std::placeholders::_1));

    //Initialise ROS timers
    update_timer_ = this->create_wall_timer(10ms, std::bind(&right_wall_follower::update_callback, this));
    RCLCPP_INFO(this->get_logger(), "Wall follower right drive node has been initialized");
  }

private:

  // ROS timer
  rclcpp::TimerBase::SharedPtr update_timer_;
```



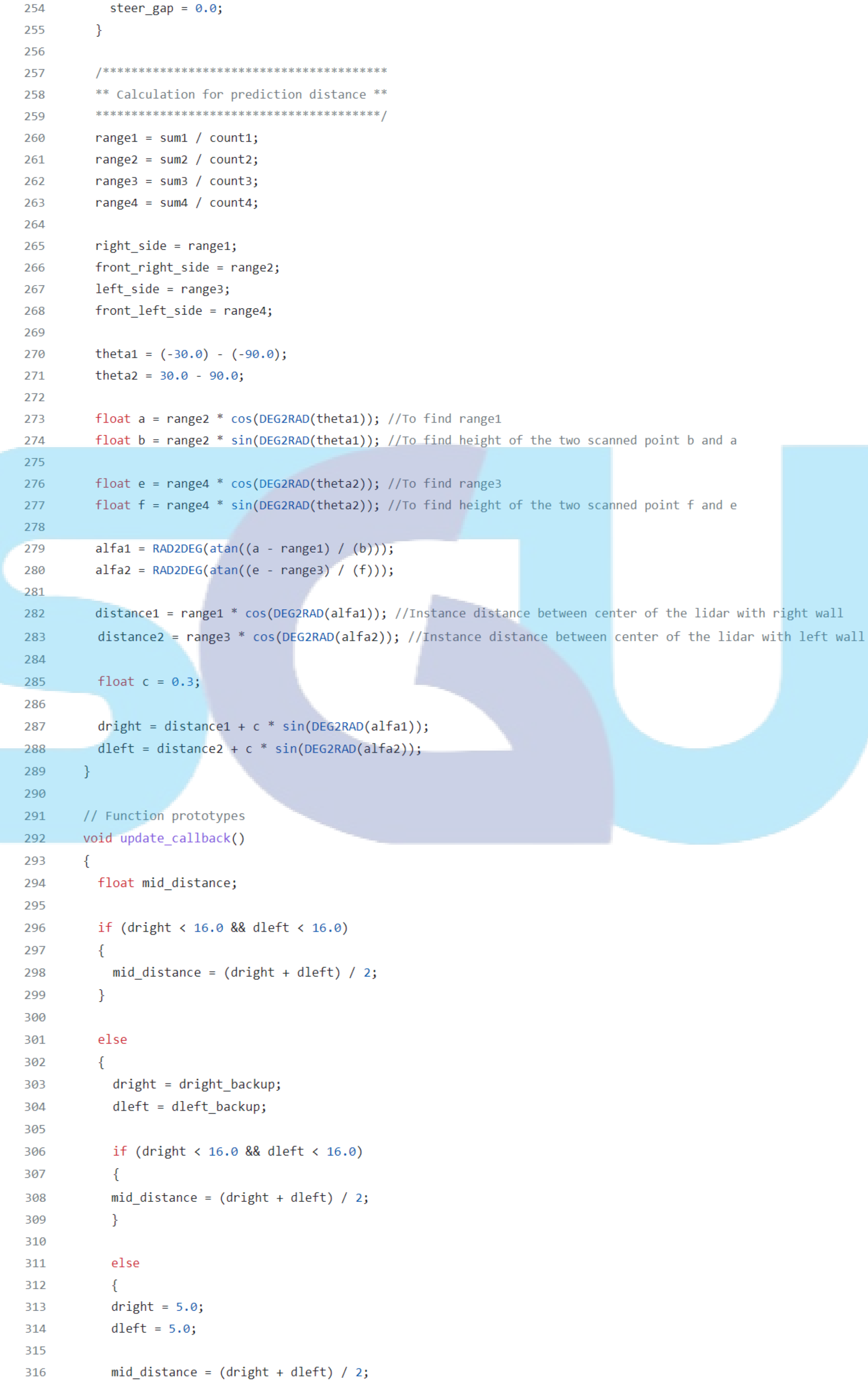
```

float right_side, left_side, front_right_side, front_left_side, theta1, theta2, kp_, ki_, kd_;
double integral_, derivative_, prev_error_;
int count1, count2, count3, count4;
float sum1, sum2, sum3, sum4, range1, range2, range3, range4, distance1, distance2, dright, dleft, alfa1, alfa2;
float last_sum1, last_sum2, last_sum3, last_sum4, last_count1, last_count2, last_count3, last_count4;
float front_view, total_left, total_right, dright_backup, dleft_backup;
double steer_gap;
float gap_index_left, gap_index_right, gap_left, gap_right, avg_index_left, avg_index_right;

void scan_callback(const sensor_msgs::msg::LaserScan::SharedPtr msg)
{
  count1 = count2 = count3 = count4 = 0;
  range1 = range2 = range3 = range4 = 0;
  sum1 = sum2 = sum3 = sum4 = 0;

  std::vector<float> laser_ranges;
  laser_ranges = msg->ranges;

  float count = msg->scan_time / msg->time_increment;

  front_view = laser_ranges[count/2];

  if (front_view > 16.0)
  {
    front_view = 16.0;
  }

  else if (front_view < 16.0)
  {
    front_view = front_view;
  }

  else
  {
    front_view = 16.0;
  }

  /* FOR LOOP*/
  for (int i = 0; i < count; i++)
  {
    float degree = RAD2DEG(msg->angle_min + msg->angle_increment * i);

    /**************************************
    ** When DEGREE is between -89 & -91 **
    **************************************/
    if (degree < -89.0 && degree > -91.0)
    {
      if (laser_ranges[i] > 0)
      {
        sum1 = sum1 + msg->ranges[i];
        count1++;
        last_sum1 = sum1;
        last_count1 = count1;
      }
      else
      {
        sum1 = last_sum1;
        count1 = last_count1;
      }
      dright_backup = laser_ranges[i];
    }

```



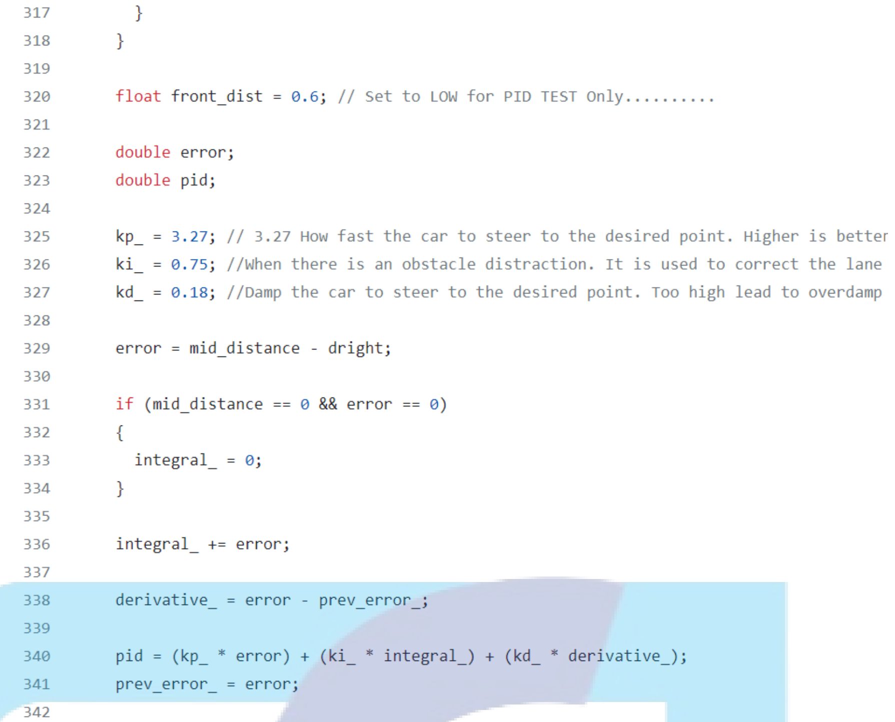
```
/*************************************
** When DEGREE is between -29 & -31 **
*************************************/
if (degree < -29.0 && degree > -31.0)
{
  if (laser_ranges[i] > 0)
  {
    sum2 = sum2 + msg->ranges[i];
    count2++;
    last_sum2 = sum2;
    last_count2 = count2;
  }
  else
  {
    sum2 = last_sum2;
    count2 = last_count2;
  }
}

/*****************************************
** When DEGREE is between 91 & 89 **
******************************************/
if (degree < 91.0 && degree > 89.0)
{
  if (laser_ranges[i] > 0)
  {
    sum3 = sum3 + msg->ranges[i];
    count3++;
    last_sum3 = sum3;
    last_count3 = count3;
  }
  else
  {
    sum3 = last_sum3;
    count3 = last_count3;
  }
  dleft_backup = laser_ranges[i];
}

/******************************************
  ** When DEGREE is between 31 & 29 **
*****************************************/
if (degree < 31.0 && degree > 29.0)
{
  if (laser_ranges[i] > 0)
  {
    sum4 = sum4 + msg->ranges[i];
    count4++;
    last_sum4 = sum4;
    last_count4 = count4;
  }
  else
  {
    sum4 = last_sum4;
    count4 = last_count4;
  }
}

/***************************************
      ** To scan from 80 to -80 **
***************************************/
if (degree > -80.0 && degree < 80.0)
{
  if (laser_ranges[i] > 16.0)
  {
```



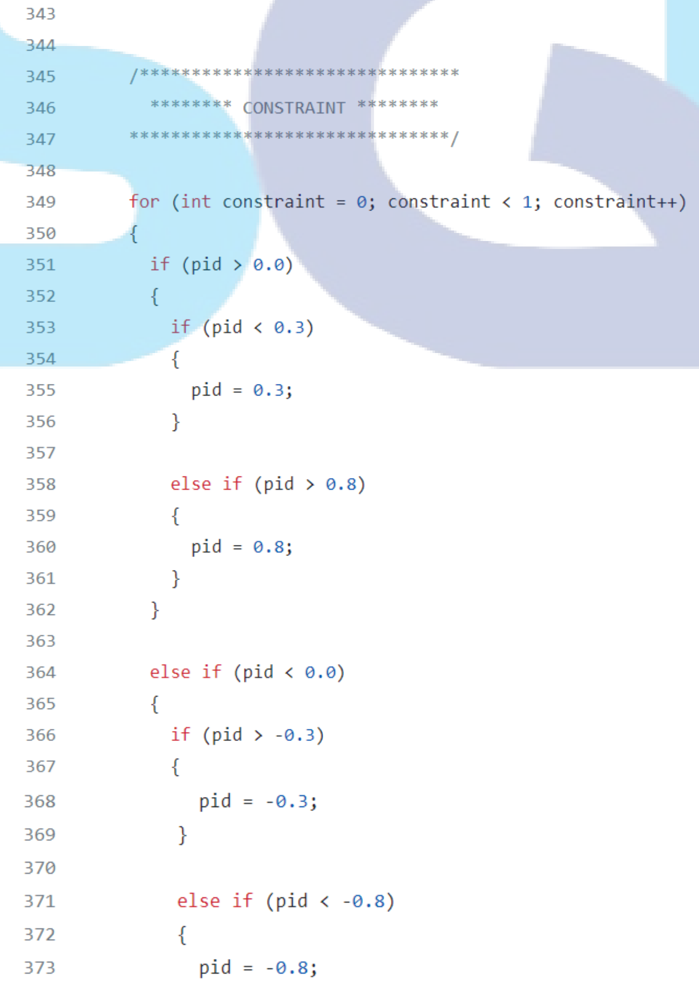
```
        laser_ranges[i] = 5.0;
      }

      if (degree > 10.0 && degree <= 60.0)
      {
        if (front_view <= 0.7) //SMALLER is better, too small makes lag response
        {
          total_left += laser_ranges[i];
        }

        else
        {
          total_left = 0.0;
        }

        if (laser_ranges[i] >= 0.7) //Value 0.7 can be adjusted. If late turn ++ if too fast
        {
          gap_index_left += i;
          gap_left++;
        }
      }

      else if (degree < -10.0 && degree >= -60.00)
      {
        if (front_view <= 0.7)
        {
          total_right += laser_ranges[i];
        }

        else
        {
          total_right = 0.0;
        }

        if (laser_ranges[i] >= 0.7)
        {
          gap_index_right += i;
          gap_right++;
        }
      }
    }
  }

  avg_index_right = gap_index_right / gap_right;
  avg_index_left = gap_index_left / gap_left;

  //printf("total RIGHT is %f & total LEFT is %f\n", total_right, total_left);

  if (total_left > total_right)
  {
    steer_gap = -1 * (msg->angle_increment * ((count/2) - avg_index_left));

    if (front_view < 0.400)
    {
      steer_gap = 0.8;
    }
  }

  else if (total_right > total_left)
  {
    steer_gap = msg->angle_increment * (avg_index_right - (count/2));

    if (front_view < 0.400)
    {
      steer_gap = -0.8;
    }
  }

  else if (total_left == total_right)
  {
```



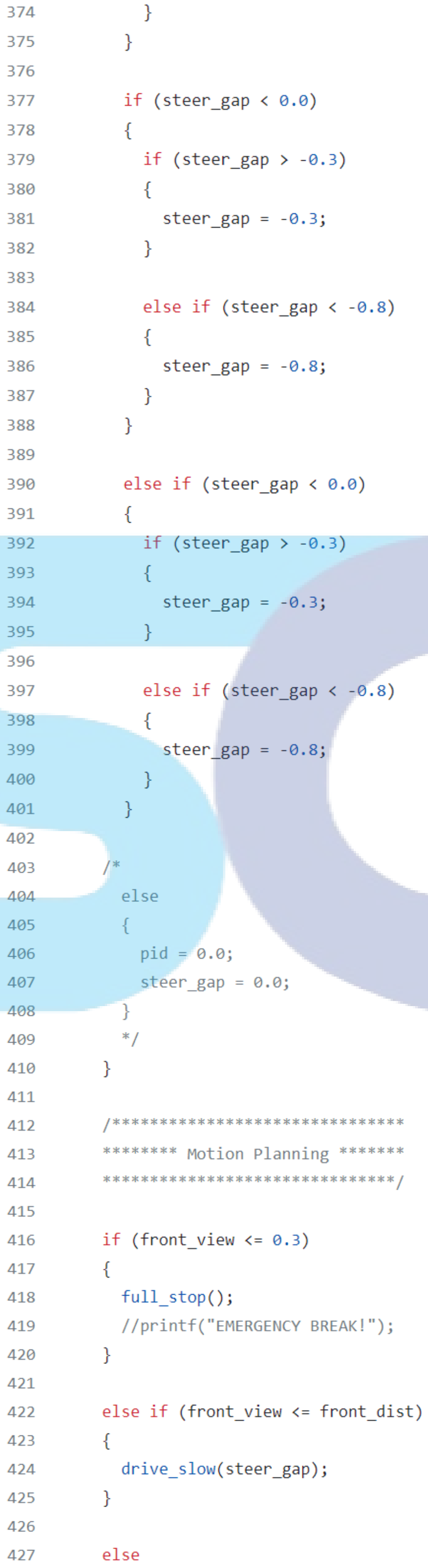
```
    steer_gap = 0.0;
  }

  /****************************************
  ** Calculation for prediction distance **
  ****************************************/
  range1 = sum1 / count1;
  range2 = sum2 / count2;
  range3 = sum3 / count3;
  range4 = sum4 / count4;

  right_side = range1;
  front_right_side = range2;
  left_side = range3;
  front_left_side = range4;

  theta1 = (-30.0) - (-90.0);
  theta2 = 30.0 - 90.0;

  float a = range2 * cos(DEG2RAD(theta1)); //To find range1
  float b = range2 * sin(DEG2RAD(theta1)); //To find height of the two scanned point b and a

  float e = range4 * cos(DEG2RAD(theta2)); //To find range3
  float f = range4 * sin(DEG2RAD(theta2)); //To find height of the two scanned point f and e

  alfa1 = RAD2DEG(atan((a - range1) / (b)));
  alfa2 = RAD2DEG(atan((e - range3) / (f)));

  distance1 = range1 * cos(DEG2RAD(alfa1)); //Instance distance between center of the lidar with right wall
  distance2 = range3 * cos(DEG2RAD(alfa2)); //Instance distance between center of the lidar with left wall

  float c = 0.3;

  dright = distance1 + c * sin(DEG2RAD(alfa1));
  dleft = distance2 + c * sin(DEG2RAD(alfa2));
}

// Function prototypes
void update_callback()
{
  float mid_distance;

  if (dright < 16.0 && dleft < 16.0)
  {
    mid_distance = (dright + dleft) / 2;
  }

  else
  {
    dright = dright_backup;
    dleft = dleft_backup;

    if (dright < 16.0 && dleft < 16.0)
    {
    mid_distance = (dright + dleft) / 2;
    }

    else
    {
    dright = 5.0;
    dleft = 5.0;

    mid_distance = (dright + dleft) / 2;
```



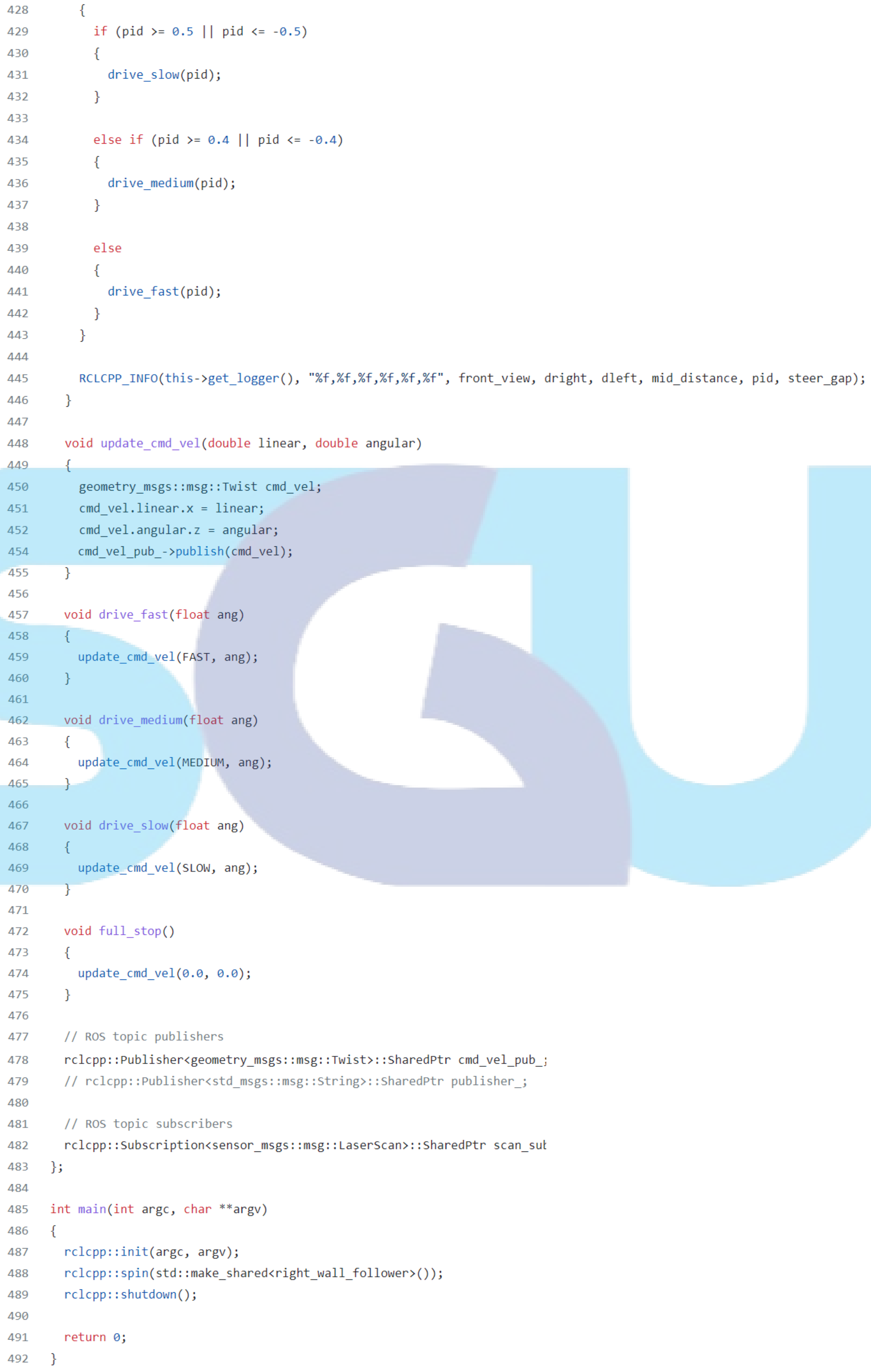
```
  }
}

float front_dist = 0.6; // Set to LOW for PID TEST Only.........

double error;
double pid;

kp_ = 3.27; // 3.27 How fast the car to steer to the desired point. Higher is better
ki_ = 0.75; //When there is an obstacle distraction. It is used to correct the lane
kd_ = 0.18; //Damp the car to steer to the desired point. Too high lead to overdamp

error = mid_distance - dright;

if (mid_distance == 0 && error == 0)
{
  integral_ = 0;
}

integral_ += error;

derivative_ = error - prev_error_;

pid = (kp_ * error) + (ki_ * integral_) + (kd_ * derivative_);
prev_error_ = error;

```



```


  /*******************************
    ******** CONSTRAINT ********
  *******************************/

  for (int constraint = 0; constraint < 1; constraint++)
  {
    if (pid > 0.0)
    {
      if (pid < 0.3)
      {
        pid = 0.3;
      }

      else if (pid > 0.8)
      {
        pid = 0.8;
      }
    }

    else if (pid < 0.0)
    {
      if (pid > -0.3)
      {
         pid = -0.3;
       }

       else if (pid < -0.8)
       {
         pid = -0.8;
```



```
      }
    }

    if (steer_gap < 0.0)
    {
      if (steer_gap > -0.3)
      {
        steer_gap = -0.3;
      }

      else if (steer_gap < -0.8)
      {
        steer_gap = -0.8;
      }
    }

    else if (steer_gap < 0.0)
    {
      if (steer_gap > -0.3)
      {
        steer_gap = -0.3;
      }

      else if (steer_gap < -0.8)
      {
        steer_gap = -0.8;
      }
    }

  /*
    else
    {
      pid = 0.0;
      steer_gap = 0.0;
    }
    */
  }

  /******************************
  ******** Motion Planning *******
  ******************************/

  if (front_view <= 0.3)
  {
    full_stop();
    //printf("EMERGENCY BREAK!");
  }

  else if (front_view <= front_dist)
  {
    drive_slow(steer_gap);
  }

  else
```

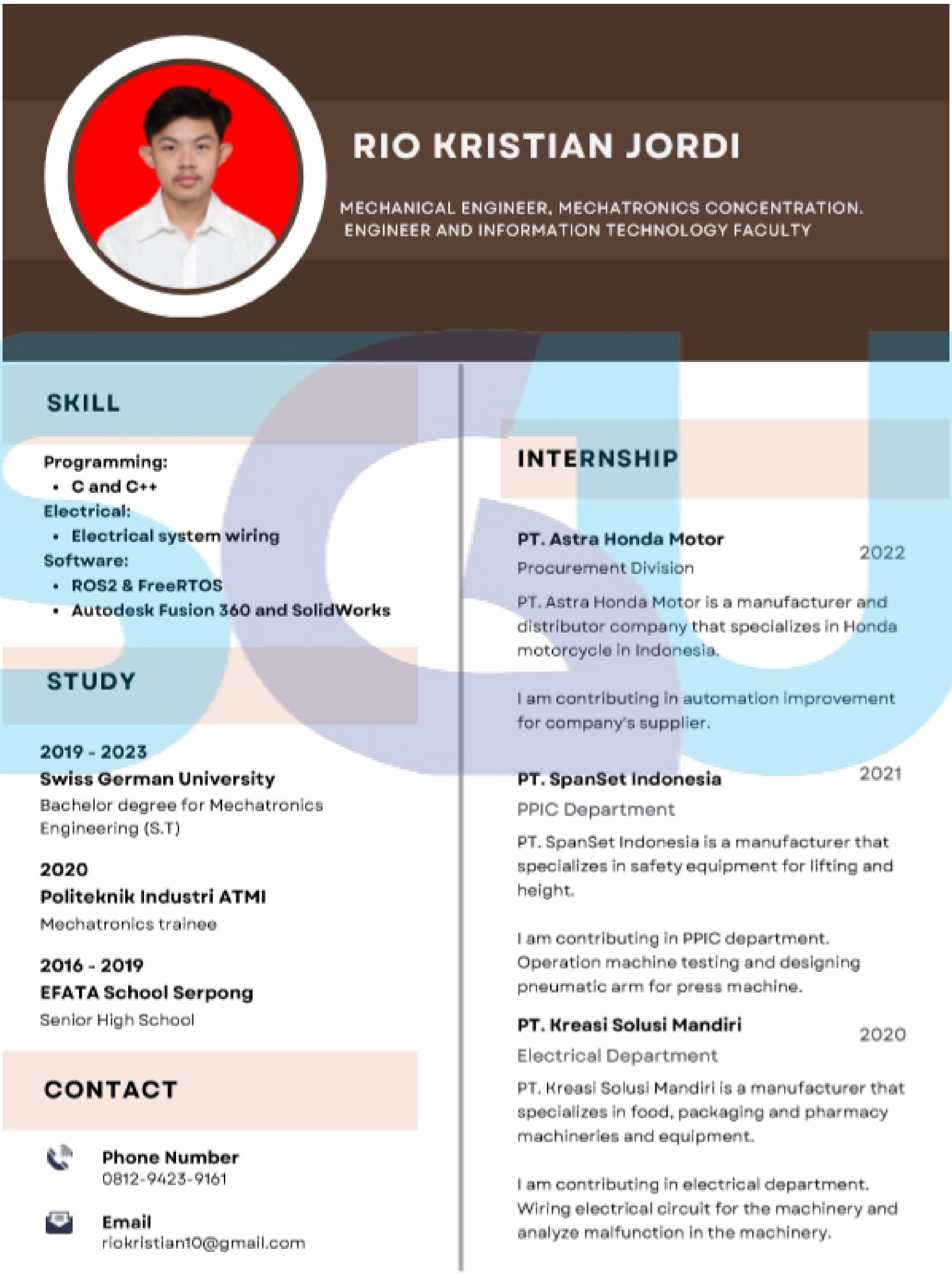


```
    {
      if (pid >= 0.5 || pid <= -0.5)
      {
        drive_slow(pid);
      }

      else if (pid >= 0.4 || pid <= -0.4)
      {
        drive_medium(pid);
      }

      else
      {
        drive_fast(pid);
      }
    }

    RCLCPP_INFO(this->get_logger(), "%f,%f,%f,%f,%f,%f", front_view, dright, dleft, mid_distance, pid, steer_gap);
  }

  void update_cmd_vel(double linear, double angular)
  {
    geometry_msgs::msg::Twist cmd_vel;
    cmd_vel.linear.x = linear;
    cmd_vel.angular.z = angular;
    cmd_vel_pub_->publish(cmd_vel);
  }

  void drive_fast(float ang)
  {
    update_cmd_vel(FAST, ang);
  }

  void drive_medium(float ang)
  {
    update_cmd_vel(MEDIUM, ang);
  }

  void drive_slow(float ang)
  {
    update_cmd_vel(SLOW, ang);
  }

  void full_stop()
  {
    update_cmd_vel(0.0, 0.0);
  }

  // ROS topic publishers
  rclcpp::Publisher<geometry_msgs::msg::Twist>::SharedPtr cmd_vel_pub_;
  // rclcpp::Publisher<std_msgs::msg::String>::SharedPtr publisher_;

  // ROS topic subscribers
  rclcpp::Subscription<sensor_msgs::msg::LaserScan>::SharedPtr scan_sut
};

int main(int argc, char **argv)
{
  rclcpp::init(argc, argv);
  rclcpp::spin(std::make_shared<right_wall_follower>());
  rclcpp::shutdown();

  return 0;
}
```

APPENDIX C – BILL OF MATERIALS

No.	Materials	Quantity (Pcs)	Total Price (Rp.)
1	SanDisk 128GB USB Flashdisk	1	202900
2	WP 1040 Brushed ESC	1	210100
3	E6001 Steering Servo	1	148900
4	ESP32-WROOM-32E	1	176000
5	Lipo 7.4V 2500 mAh Battery	1	235800
6	IMAX B3 Pro Lipo Balance Charger	1	52300
7	Vention HDMI to Micro HDMI Cable	1	71560
8	Vention Adapter HDMI Female	1	12200
9	Small Breadboard Mini	1	13200
10	KY033 Infrared Sensor	1	57400
11	TB-1503 Terminal Block	1	12600
12	3M Bolt and Nut	5	2000
Total Price			**1194960**

CURRICULUM VITAE



RIO KRISTIAN JORDI

MECHANICAL ENGINEER, MECHATRONICS CONCENTRATION.
ENGINEER AND INFORMATION TECHNOLOGY FACULTY

SKILL

Programming:

- **C and C++**

Electrical:

- **Electrical system wiring**

Software:

- **ROS2 & FreeRTOS**
- **Autodesk Fusion 360 and SolidWorks**

STUDY

2019 - 2023
Swiss German University
Bachelor degree for Mechatronics Engineering (S.T)

2020
Politeknik Industri ATMI
Mechatronics trainee

2016 - 2019
EFATA School Serpong
Senior High School

CONTACT

☎ **Phone Number**
0812-9423-9161

✉ **Email**
riokristian10@gmail.com

INTERNSHIP

PT. Astra Honda Motor 2022
Procurement Division

PT. Astra Honda Motor is a manufacturer and distributor company that specializes in Honda motorcycle in Indonesia.

I am contributing in automation improvement for company's supplier.

PT. SpanSet Indonesia 2021
PPIC Department

PT. SpanSet Indonesia is a manufacturer that specializes in safety equipment for lifting and height.

I am contributing in PPIC department. Operation machine testing and designing pneumatic arm for press machine.

PT. Kreasi Solusi Mandiri 2020
Electrical Department

PT. Kreasi Solusi Mandiri is a manufacturer that specializes in food, packaging and pharmacy machineries and equipment.

I am contributing in electrical department. Wiring electrical circuit for the machinery and analyze malfunction in the machinery.