*Forest Chemicals Review*
*www.forestchemicalsreview.com*
*ISSN: 1520-0191*
*May-June 2022 Page No. 23-44*
*Article History: Received: 24 February 2022, Revised: 05 April 2022, Accepted: 08 May 2022, Publication: 30 June 2022*

# A Prototype of Visual Navigation Self-driving Car Architecture Based on EAIDK310 and STM32

**Xiaojing Wen[1], Wenchao Han[1], Junli Chen[2], Wenping Jiang[1*], Zhaohong Ding[1*]**
[1]School of Electrical and Electronic Engineering, Shanghai Institute of Technology, 201416, China
[2]Electrical and Computer Engineering, National University of Singapore, Singapore
*Corresponding Author.

***Abstract:***

To further understand the AI visual navigation technology, this paper design an AI car by combining STM32 MCU and EAIDK310 developed by OPENAILAB. For the hardware architecture of the intelligent car system, the EAIDK310 development kit equipped with RK3228 is used as the core processor responsible for real-time data acquisition, visual recognition, navigation control, and the STM32F405RGT6 MCU acts as the auxiliary processor to finish IMU data acquisition and instruction execution output. Meanwhile, the dual-camera scheme is adopted to accomplish the automatic driving functioned with road segmentation and traffic sign identification; one camera is used to collect real-time road information at a near distance, and the other one plays the role of capturing environmental information at a wide-angle. For the software system, an efficient, economical, and low-coupling framework is constructed by Redis and Tengine edge inference computing which acting as the critical of data interactive transmission and visual recognition processing, respectively, with this framework, the improvement of real-time and calculating the speed of the system has been realized. Finally, the operation stability of hardware and the software real-time are further proved by experiment measurements, indicating that the proposed method is a potential candidate for a low-cost AI visual navigation system.

***Keywords****: Visual navigation, Architecture, Self-driving car, Embedded system.*

## I. INTRODUCTION

The related technology about self-driving gradually enters the view of people as the self-driving deepens into human life. As one of the branches, visual navigation, facing the challenge of high requirements for computing ability, which also makes self-driving and related technology facing a bottleneck and technical threshold. Thus, the research topic focusing on optimizing the computing ability of visual navigation has developed as a hotpot for many scholars[1].

Aiming to deepen understanding of the visual navigation of self-driving algorithms, a visual-based intelligent car platform equipped with the EAIDK310 development board and STM32 MCU has been put

*Forest Chemicals Review*
*www.forestchemicalsreview.com*
*ISSN: 1520-0191*
*May-June 2022 Page No. 23-44*
*Article History: Received: 24 February 2022, Revised: 05 April 2022, Accepted: 08 May 2022, Publication: 30 June 2022*

forward, at the same time, two cameras have been installed in a specific position to serve as the sensor for inputting external information. Thus, a complete multifunction visual navigation testing platform has been set up. Furthermore, for maximizing the characteristic function of selected EAIDK310, a cost-saving, efficient, resourcing-saving software architecture has been adopted. Finally, a series of verification experiments have been implemented to evaluating the effectiveness of the self-driving platform. The experimental results proved that the proposed designing scheme is highly feasible in embedded artificial intelligence (AI) with the advantage of low cost. All the conclusions suggests that the as-designed visual-based intelligent car platform has great potential in realizing self-driving research and is helpful for next-generation intelligent applications.

## II. RELATED WORKS

### 2.1 Hardware Architecture of the Robot

On the basis of the number of processors, hardware architecture can mainly divide into single processor architecture and the architecture composed by the main processor and auxiliary processor. For the former architecture, it can be further categorized into micro control unit (MCU)-based and industrial control computer(ICC)-based via different application scenarios; the MCU is suitable for the simple task scene，while the ICC is performed better in a complex task with tedious calculations, such as the robot design scheme proposed by Du et al. [2]. In a dual-processor architecture[3-5], the main processor is used to run an operating system based on Linux with different distributions, and the auxiliary processor is served by an MCU running in a real-time operating system (ROS) is utilized to deal with real-time tasks caused by the non-real-time of the system. Moreover, the effectiveness of the dual-processor scheme has been validated by numerous projects and product applications, and the auxiliary processor can act as a remedy to compensate for the non-real-time and lacking peripheral interfaces (such as SPI，IIC，CAN, etc.) of the main processor. Herein, a similar architecture has been utilized in this paper, considering it can flexibly adjust the coordination of the main and auxiliary processors to fulfill the requirement of the different scenes within the same architecture, which is also helpful for reducing the hardware costs.

### 2.2 Software Framework of Robot

Recently, the ROS operating system has been widely used in the main processor port and served as software architecture based on the aforementioned hardware architecture. On the other hand, with the development of technology, the emergence of upgraded ROS2 has also made this type of software architecture been accepted and applied by scientists and enterprises [6-11]. Meanwhile, more and more effort has been devoted to the application of the ROS system. For example, Li et al. [12] developed a self-moving robot on the ROS-based system, and Chaudhry et al.[13] proposed an automated robotic arm system based on ROS2. However, considering the limit of non-real-time of the adopted remote procedure call (RPC) which is realized on the general-purpose operating systems (GPOS) and the incompatibility problems result from the ROS and ROS2 in the specific platform. Researchers began to shift their attention to the other frameworks. Pan et al.[14] come up with a real-time RTRG-RPC architecture to extend the robot

*Forest Chemicals Review*
*www.forestchemicalsreview.com*
*ISSN: 1520-0191*
*May-June 2022 Page No. 23-44*
*Article History: Received: 24 February 2022, Revised: 05 April 2022, Accepted: 08 May 2022, Publication: 30 June 2022*

application. Thus, we innovatively proposed a robot software architecture depending on the Redis real-time cache library, where the data interaction between nodes and nodes is finished via Redis.

2.3 The Navigation Sensor

The self-driving navigation sensor can be divided into two categories: visual and laser radar which can be used in a separate form or a hybrid form, such as the self-driving scheme using in the Tesla's electric car is achieved environmental perception through the data acquired by eight cameras, and the widely used sweeping robot is mainly completed environmental perception and navigation based on a single-line laser radar, while Aufrere [15] constructed a self-driving scheme of Jeep by combining multiple radars and cameras. However, although the perception scheme realized by a laser radar is better than a camera-based one from the software perspective, the camera-based perception scheme performs well in terms of hardware. Moreover, with the number of laser lines increasing, the rising expense also limits the application of laser radar. Also, 80% to 90% of external information is obtained from people's eyes [16]. Thus, take account of the cost consumption, the cameras were selected in our self-driving car.

2.4 Main Processor

The requirement of visual navigation for computing ability has become a challenge in the last few years, which is also a threshold for further conducting studies about visual navigation. With the development of AI technology and the launching of AI chips equipped with hardware acceleration units, a significant breakthrough has been made in the field of chips, such as the typical K210 MCU proposed by Jiannan technology with the computer ability of 1TOPS which is suitable for high computer AI scenes[17]. Besides, Nvidia invented Jeston series modules (Nano, tx2, Xaviar, etc.) which all belong to embedded system platforms with GPU acceleration units [18,19]. All of these progress path a way for utilizing visual navigation into actual products.

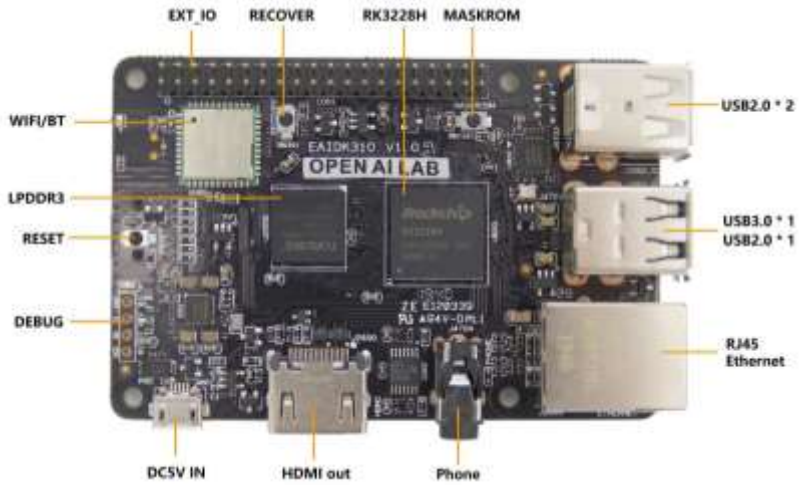


Fig 1: Real photo of EAIDK310

The Embedded Artificial Intelligence Development Kit 310 (EAIDK310) used in this paper is an

*Forest Chemicals Review*
*www.forestchemicalsreview.com*
*ISSN: 1520-0191*
*May-June 2022 Page No. 23-44*
*Article History: Received: 24 February 2022, Revised: 05 April 2022, Accepted: 08 May 2022, Publication: 30 June 2022*

embedded AI development platform developed by OPENAILAB for edge computing, and it is equipped with the AI development platform including the deep learning framework of Tengine and the lightweight embedded computer vision acceleration library of BladeCV[20]. Moreover, this development board supports the AI application by providing a simple, efficient, and unified API interface and accelerates the implementation of AI terminal products for specific scenarios. The specification and appearance of the EAIDK310 in this paper are listed in Table A.1, Table A.2, and Fig 1, respectively.

## III. Designing method

### 3.1 System Framework Designing

#### 3.1.1 System architecture

The overall system structure of our design is shown in Fig 2. It mainly includes the hardware layer, driver layer, data interaction layer, data processing layer, control algorithm layer and application layer from bottom to top.

The hardware layer is the most intuitive physical embodiment of the self-driving car design, mainly composed of motors, servos, speed sensors, IMU sensors, button buzzers, etc.[21]

The driver layer is the lowest level software to directly interacts with the hardware and driving it runs, including the PWM that drives the motor servo, the input capture of the speed sensor, the IIC bus for driving the IMU, as well as the GPIO and UART for human-machine and machine-machine interaction.

The data interaction layer is the critical layer that connects the bottom layer and the upper layer. It is an abstraction and encapsulation to the driver layer from the perspective of data flow. Also, it supplies a unified interface and permission control for accessing and controlling the upper layer applications to the underlying hardware.

The data operation layer is the first layer of data processing, which mainly provides the most basic mathematical operation and transformation as well as the methods for digital signal processing. Then, based on the origin or processed data proving by the data operation layer, the control algorithm layer realizes the control of the model by bringing the data into the actual model.

*Forest Chemicals Review*
*www.forestchemicalsreview.com*
*ISSN: 1520-0191*
*May-June 2022 Page No. 23-44*
*Article History: Received: 24 February 2022, Revised: 05 April 2022, Accepted: 08 May 2022, Publication: 30 June 2022*

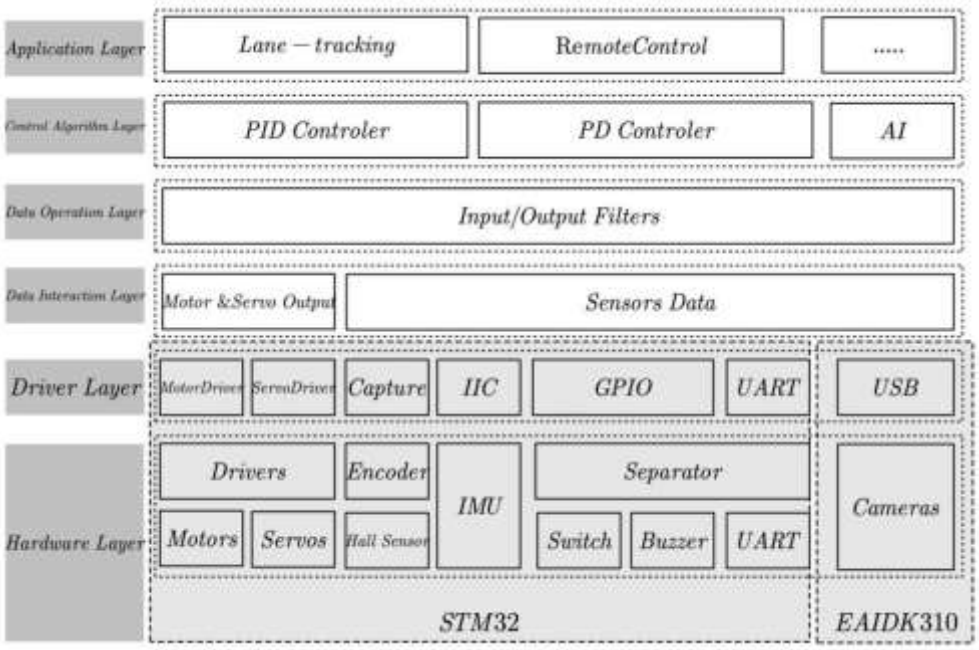


Fig 2: The diagram of system architecture

As the uppermost logical processing layer, the application layer is utilized to execute the specific function, such as making assembled car lane tracking or controlling the driving strategy of the car.

3.1.2 Hardware architecture of the system

The hardware frame of the designed system is diagramed in Fig 3; it consists of the main controller and the auxiliary controller. The main controller is the EAIDK310 development board load with the RK3228H processor and is responsible for the acquisition and processing of the camera data as well as executing the main control algorithm. The auxiliary processor communicating with the main controller via a serial port is a 32-bit STM32F405 MCU, which is mainly used for driving the motor, controlling speed and direction, and calculating the posture of the car. Especially, a dual-camera scheme (Fig 4(a)) is adopted on the side of the main processor, one camera which is positioned at the height of ~15 cm to the ending of the car with an installation angle of 12~15°, is used to collect the position of the car on the road, and the view scope is approximate twice the length of the car, the other micro-wide-angle camera installing in the front of the car is mainly to collect real-time road condition information such as pedestrian traffic signs, the view of the cameras are shown in Fig 4(b). Compared with the single-camera scheme, the dual-camera is efficient for reducing the preprocessing of the image by the main processor and auxiliary processor since the data collected by the dual-camera can be directly used for road recognition, segmentation, and traffic sign processing.

*Forest Chemicals Review*
*www.forestchemicalsreview.com*
*ISSN: 1520-0191*
*May-June 2022 Page No. 23-44*
*Article History: Received: 24 February 2022, Revised: 05 April 2022, Accepted: 08 May 2022, Publication: 30 June 2022*

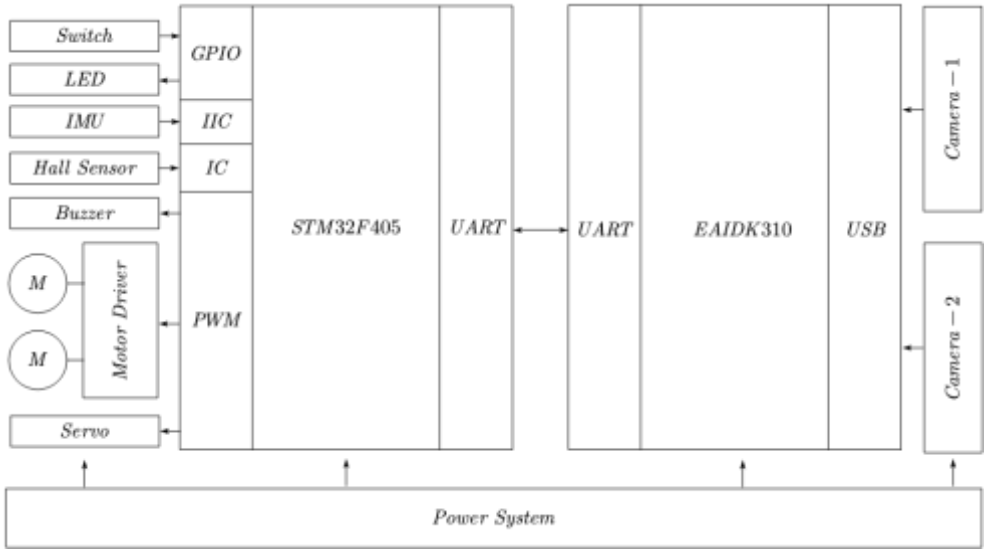


Fig 3: The diagram of hardware architecture

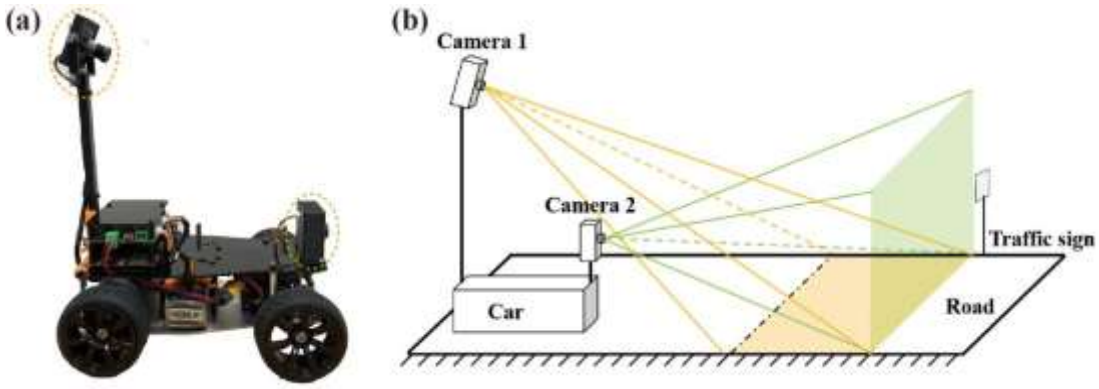


Fig 4: Schematic diagram of the dual-camera scheme (a) the actual installation position of the cameras; (b) the view of the camera.

3.1.3 Software architecture of the system

Corresponding to the hardware system, the software architecture can also be divided into the primary processor side and the auxiliary processor side. As shown in Fig 5, OpenCV is applied in the main processor to accomplish the image acquisition and fundamental processing work, meanwhile, using Tengine to implement the inference calculation of the artificial intelligence training model, and the data processed and calculated by the above software are written into Redis to prepare for further calls of other programs (vehicle control algorithm, data protocol, image recognition modules, etc.). The car control driver is a node written in python that communicates with the STM32 MCU by serial port. It real-time reads the control instructions of the car from the upper application (vehicle control algorithm) in Redis and then sends it to the auxiliary processor through the serial port protocol. Finally, The car makes

*Forest Chemicals Review*
*www.forestchemicalsreview.com*
*ISSN: 1520-0191*
*May-June 2022 Page No. 23-44*
*Article History: Received: 24 February 2022, Revised: 05 April 2022, Accepted: 08 May 2022, Publication: 30 June 2022*

corresponding actions according to the instructions.

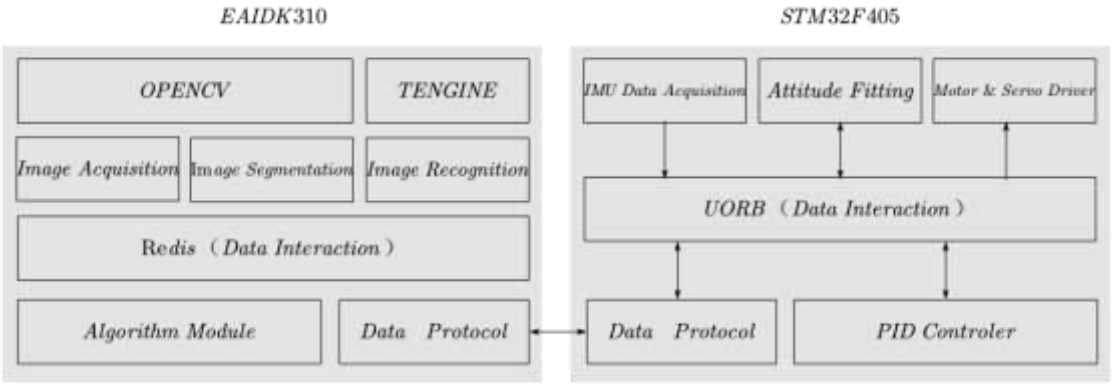


Fig 5 The diagram of software architecture

3.2 The Circuit Designing of Hardware

3.2.1 STM32 minimum system board

The minimum system of the car system board is shown in Fig A.1. The MCU used in the design is STM32F405RGT6 with an external clock source of 8Mhz, and the digital power (VDD) of the MCU and the analog power supply (VDD_A) are connected by using an inductor (L1) of 1mH to improve the power stability and the anti-interference of the MCU.

3.2.2 Power supply system

Considering the general power input is DC12V, while the voltages required in this system are DC5V and DC3.3V. Thus, the power supply system is designed to finish the conversion of the voltage. As can be seen in Fig 6, the step-down DC-DC convertor (SY8105) capable of delivering a 5.0A current is adopted in this system based on its wide input voltage range from 4.5V to 18V, which is sufficient for converting DC12V to DC5V. As for the DC3.3V plays the role of supplying power for MCU and parts of driving chips. On the one hand, on account of the operational stability of the MCU, we use a linear regulator chip with the model TPS73633; on the other hand, to reduce the power consumption of the linear regulator chip, the voltage switching form is achieved by converting 5V to 3.3V.

In addition, the voltage monitoring circuit and the current monitoring circuit are also constructed to better monitor and manage the power supply system of the car body. The circuit principle schematic of this system is plotting in Fig 7. Among this system, the real-time voltages are obtained by measuring the sub-voltage of R30 and R31 resistors arranging in a series form, the current monitoring is done by using the INA169 chip to measure the voltage across the resistor R20, and then the current value can be calculated according to the Ohm's law.

*Forest Chemicals Review*
*www.forestchemicalsreview.com*
*ISSN: 1520-0191*
*May-June 2022 Page No. 23-44*
*Article History: Received: 24 February 2022, Revised: 05 April 2022, Accepted: 08 May 2022, Publication: 30 June 2022*

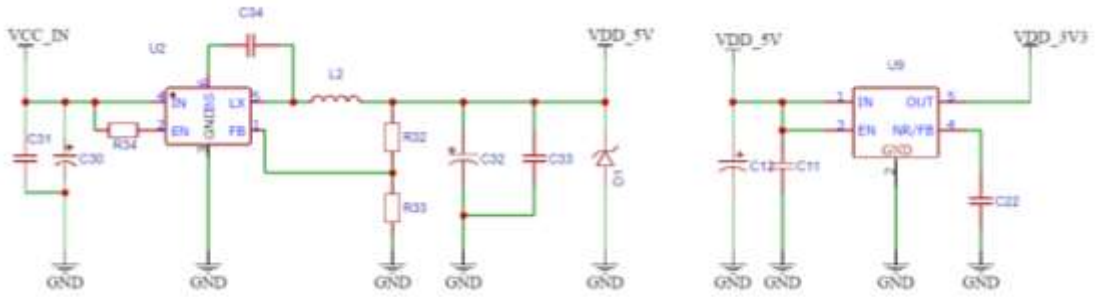


Fig 6: Circuit principal diagram of the power supply system

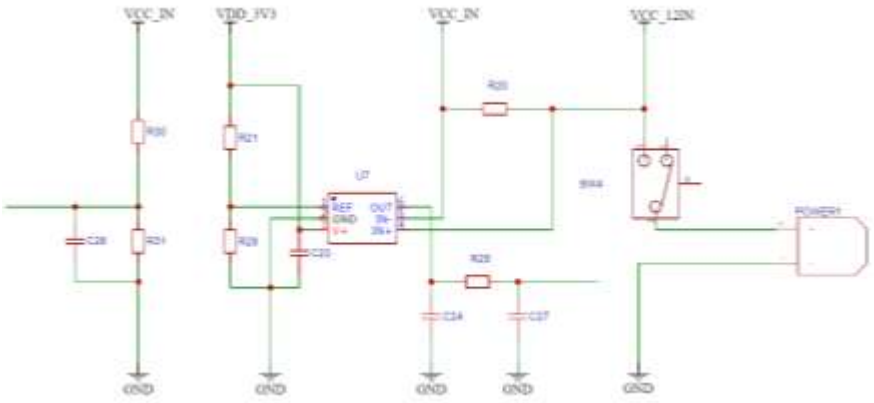


Fig 7: Circuit principal diagram of the power supply monitoring system

3.2.3 Motor driving system and IMU system

For the motor driving system, the motor model used here is a JGB37-520 DC geared motor, with a rated voltage of 12V, a no-load current of 0.07A, a rated current of 0.35A, and a locked-rotor current of 2.8A. Hence, The AT8870 chip (voltage range: 6.5V~38V, peak drive current output: 3.6A, continuous output capacity: 2A) is chosen here to leave enough current margin for the system to operate safely and securely operate state. The circuit schematic diagram of this system is shown in Fig 8.

*Forest Chemicals Review*
*www.forestchemicalsreview.com*
*ISSN: 1520-0191*
*May-June 2022 Page No. 23-44*
*Article History: Received: 24 February 2022, Revised: 05 April 2022, Accepted: 08 May 2022, Publication: 30 June 2022*

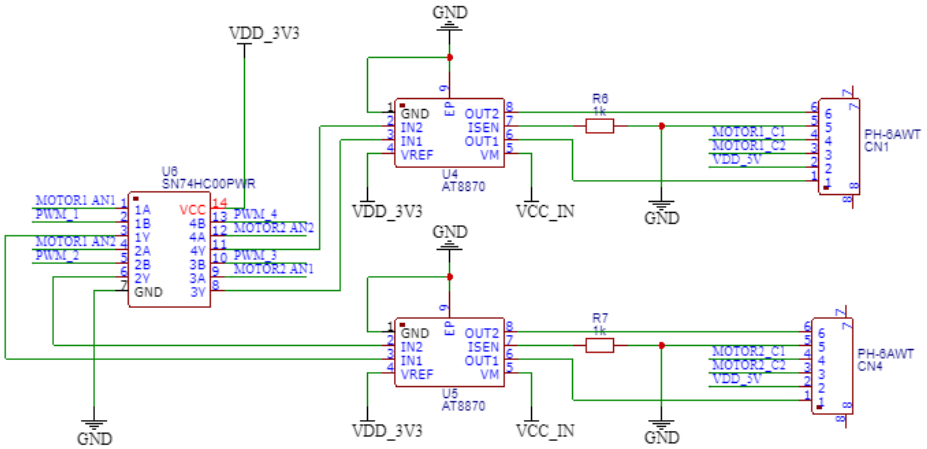


Fig 8: Circuit principal diagram of the motor driving system

For the IMU system, The circuit schematic of the IMU sensor is shown in Fig 9. The MEMs sensor (MPU9250) is equipped and the communication method is IIC.

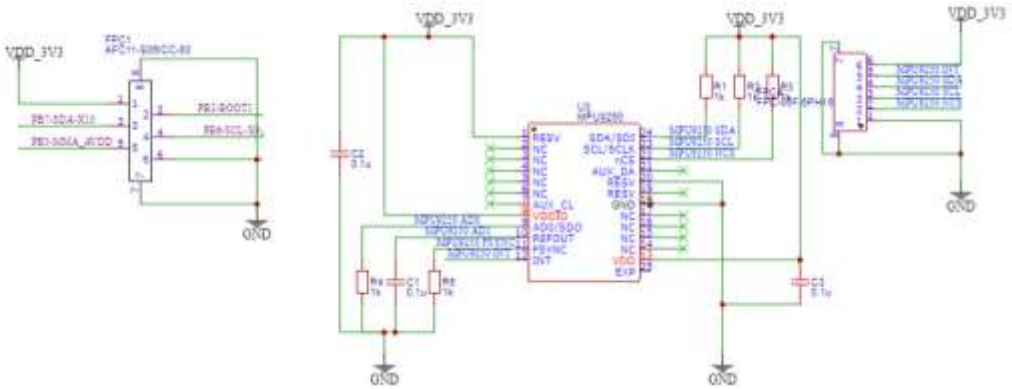


Fig 9: Circuit principal diagram of the IMU system

3.2.4 Interface and indicator system

The interface and principle of the system board are depicted in Fig 10. A series of interfaces covering servo, indicator light, TF card, USB and so on are designed. Besides, aiming to improve the load capacity of the MCU and isolate the external signals, the SN74HC245PWR chip is arranged in the part of servo and indicator.

*Forest Chemicals Review*
*www.forestchemicalsreview.com*
*ISSN: 1520-0191*
*May-June 2022 Page No. 23-44*
*Article History: Received: 24 February 2022, Revised: 05 April 2022, Accepted: 08 May 2022, Publication: 30 June 2022*

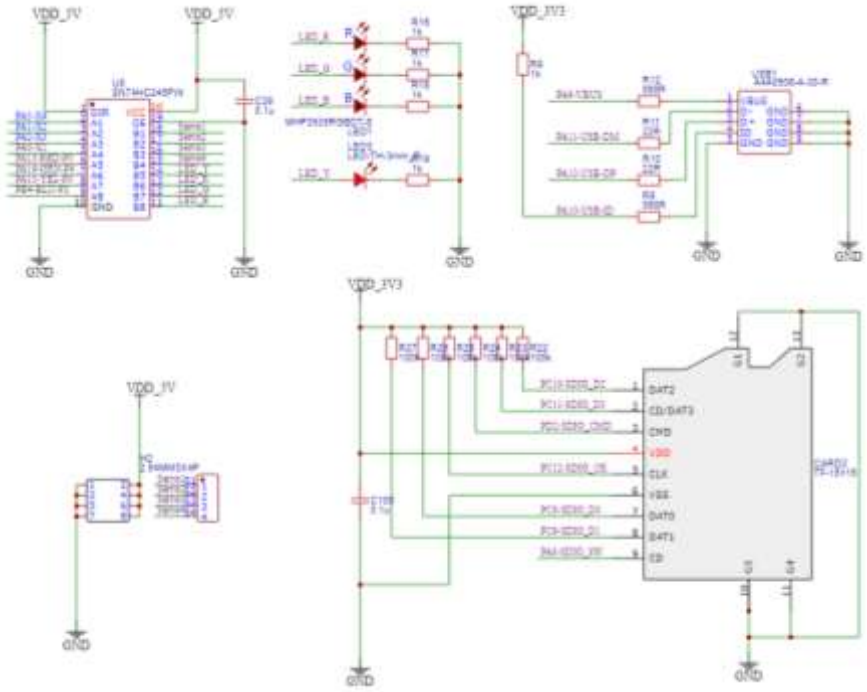

Fig 10: Circuit principal diagram of the interface and indicator system

3.3 The Designing of the Software System

3.3.1 Designing of the data protocol

In order to realize the regular communication between the main processor and the auxiliary processor, it is necessary to stipulate the data protocol that follows by both parties, the corresponding data structures of the protocol based on a bytecode format are given in Fig 11. It can be seen that the designed data protocol mainly consists of seven parts. Firstly, the frame header with fixed content is focused on identifying the start position of the data. Secondly, The LEN and SEQ described in Fig 11 refer to the effective data length of the payload and a number that is cyclically incremented from 0 to 255 to determine the number of lost packets, respectively. Meanwhile, the SYS_ID is used to distinguish different devices. For example, this paper's SYS_ID of the MCU and the EAIDK310 are 0x32 and 0x31, respectively, yet the MSG_ID guides payload for data analysis. Finally, the last two bytes of CRC (16-bit) verifying the correctness of the received data. In words, the designed protocol is efficient without excessive redundant bytes. Simultaneously, the realization of supplementing and extending this protocol can be done just by extending MSG_ID.

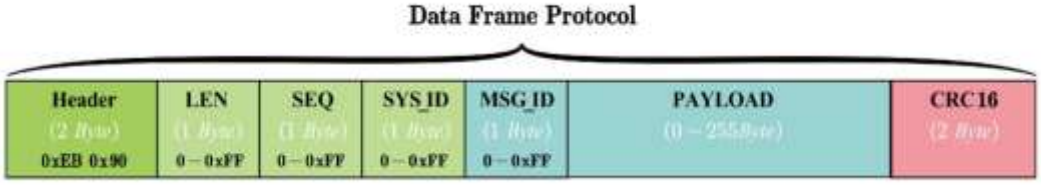


Fig 11: The structure diagram of the data protocol

*Forest Chemicals Review*
*www.forestchemicalsreview.com*
*ISSN: 1520-0191*
*May-June 2022 Page No. 23-44*
*Article History: Received: 24 February 2022, Revised: 05 April 2022, Accepted: 08 May 2022, Publication: 30 June 2022*

3.3.2 Software designing of the STM32 MCU

The software flow chart of the STM32 MCU terminal is shown in Fig 12.

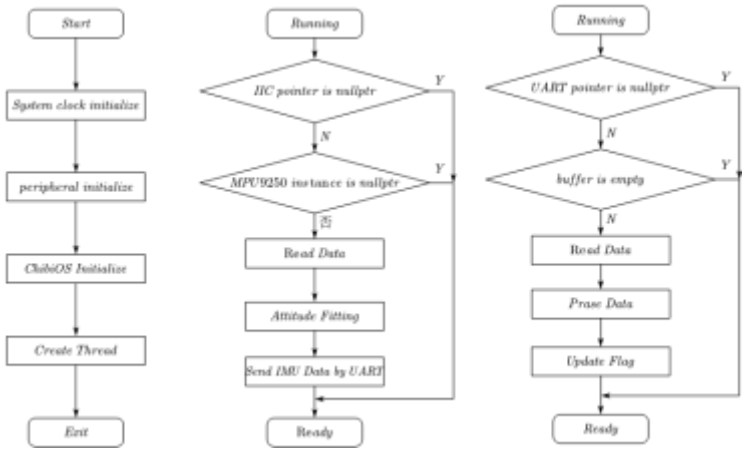


Fig 12 Flow chart of the STM32F405RG program

3.3.3 Software designing of EAIDK310

Camera data acquisition software: The collection of real-time data of the self-driving car is mainly achieved by camera data acquisition software. Taking into account the difference in video numbers caused by the randomness of the order in which the two cameras are loaded into the Dev directory, the following method has been developed to overcome this problem: the parameters of the cameras have been read during initialization, followed by a registration mechanism to conduct the register for the initialized cameras, and finally the application layer distinguished the front and the rear cameras according to the corresponding registry information. The process of the designed camera data acquisition program can be seen in Fig 13.

*Forest Chemicals Review*
*www.forestchemicalsreview.com*
*ISSN: 1520-0191*
*May-June 2022 Page No. 23-44*
*Article History: Received: 24 February 2022, Revised: 05 April 2022, Accepted: 08 May 2022, Publication: 30 June 2022*

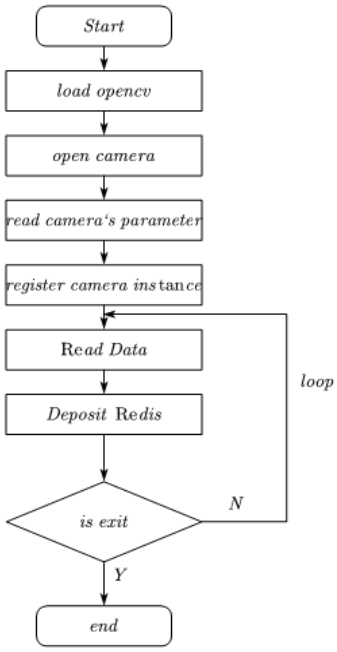


Fig 13: Flow chart of the camera data acquisition program

Functional evaluation and algorithm of the self-driving car: Herein, the road segment and the traffic signs recognition function have been used to verify and evaluate the self-driving car's function, the machine vision methods based on an OpenCV framework are used to segment and recognize the road. Thus the self-driving car can be driving with a line patrol function. In comparison, the recognition of traffic signs is mainly via an AI algorithm.

From the view of the algorithm, the algorithm adopted in the road segmentation is the fusion of edge detection, HSV color detection, and straight-line detection. Besides, basic operations such as expansion and corrosion are used to filter out the noise that appeared during the recognition processing [22-24]. And the SSD model classification and detection algorithm based on the RFB network is using in the traffic sign recognition [25,26].

3.3.4 Software design of serial communication

The serial port is the channel between the main controller and the auxiliary controller. The main controller reads the status data from the auxiliary controller and executes the feedback data of the command via serial port. On the contrary, the auxiliary controller reads the command information from the main controller in real-time through the serial port. The way to decode the data received by the processors is the state mechanism. The whole decoding process can be divided into three states: receiving frame header state, receiving data state, and receiving CRC check state. The receiving flow chart is shown in Fig A.2. In addition, in order to efficiently and quickly perform encoding and decoding operations, union packed is used to compress and align the data based on the memory while programming. When encoding, the value can be assigned to the data according to the order of the struct, so the serial port can directly read

*Forest Chemicals Review*
*www.forestchemicalsreview.com*
*ISSN: 1520-0191*
*May-June 2022 Page No. 23-44*
*Article History: Received: 24 February 2022, Revised: 05 April 2022, Accepted: 08 May 2022, Publication: 30 June 2022*

the byte of the union packed and send it out. Then in the process of decoding, the data is sequentially placed into the byte of the union after finding the frame header. Finally, the value of the command or the parameter can be obtained from the struct. To sum, this software design of serial communication effectively boosts the efficiency of coding and decoding without additional conversion processes. Herein, the IMU data of the designed car is taken as an example, and the corresponding codes are shown in Fig A.3.

3.3.5 Monitoring and debugging software of the system

The remote monitoring and debugging software is designed based on the PyQt5 framework to further simplify the real-time monitor process of the self-driving car. The interface of as-designed software is plotted in Fig 14. Within the software, The connection between the PC and the car is set up by the internet, and the real-time information transmission is realized via establishing a connection between the vehicle-mounted Redis server and the remote control commands as well as the car status. Two methods can be adopted for the video information acquiring: one is to complete the video conversion by obtaining the picture information in the vehicle-mounted Redis server, and the other is to obtain the real-time video from the camera in the vehicle-mounted RTMP server in the way of RTMP streaming. Also, Since the PyQt is cross-platform, thus the software can run on different platforms, including the car platform.

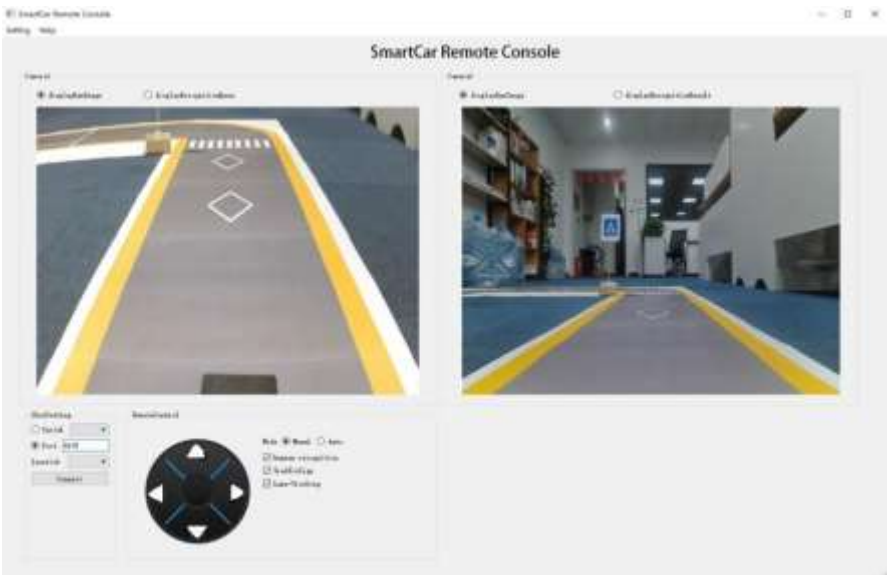


Fig 14: System interface diagram of the monitoring and debugging software

## Ⅳ. EVALUATION AND DISCUSSION

The mechanical design and actual photo of the designed intelligent car are shown in Fig 15. The respective testing on the car, including remote driving, road segmentation, traffic signs recognition, etc., all

*Forest Chemicals Review*
*www.forestchemicalsreview.com*
*ISSN: 1520-0191*
*May-June 2022 Page No. 23-44*
*Article History: Received: 24 February 2022, Revised: 05 April 2022, Accepted: 08 May 2022, Publication: 30 June 2022*

the measurement results proved that the as-proposed visual navigation self-driving car platform is stable to satisfy further multi-functional development and research.

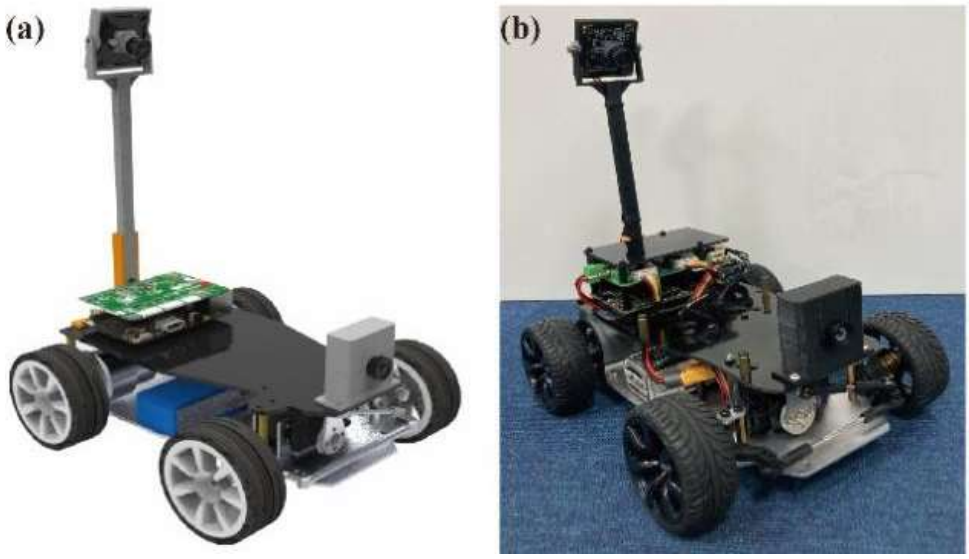


Fig 15 (a) Mechanical structure and (b) real photo of the self-driving car

4.1 Remote Control Driving Testing

It can be concluded that general controls such as forward, backward, turning left, and turning right of the car can be achieved by dragging the virtual joystick on the remote monitoring software. At the same time, the car can be controlled to drive at a certain speed and turn radius according to the position of the virtual joystick.

4.2 Camera Data Acquisition Testing

For demonstrating the data collection function and the operating performance of the cameras, three experiments were developed, the delay of the data in every test was also recorded and analyzed.

Experiment 1: Connect the EAIDK310 development board to the monitor through the HDMI cable. So the monitor can directly display the real-time images collected by the camera. The measured data in this control group can be used as the limit value for the development board collecting and processing images.

Experiment 2: The collected data are first written into Redis through the data collection node of the camera, and then subscribe to the node and connect the car to the Redis server through 127.0.0.1 (Experiment 2-a) and LAN IP (Experiment 2-b) to further read image data and real-time screen display.

Experiment 3: The data collected from the nodes is firstly published on the server through RTMP video

*Forest Chemicals Review*
*www.forestchemicalsreview.com*
*ISSN: 1520-0191*
*May-June 2022 Page No. 23-44*
*Article History: Received: 24 February 2022, Revised: 05 April 2022, Accepted: 08 May 2022, Publication: 30 June 2022*

streaming, then the remote monitoring software is connected to the RTMP server via 127.0.0.1 (Experiment 3-a) and the LAN IP of the car (Experiment 3-b) to observe the real-time images.

It should be noted that all unnecessary programs in the experiments have been closed, and only the acquisition and display programs are run.

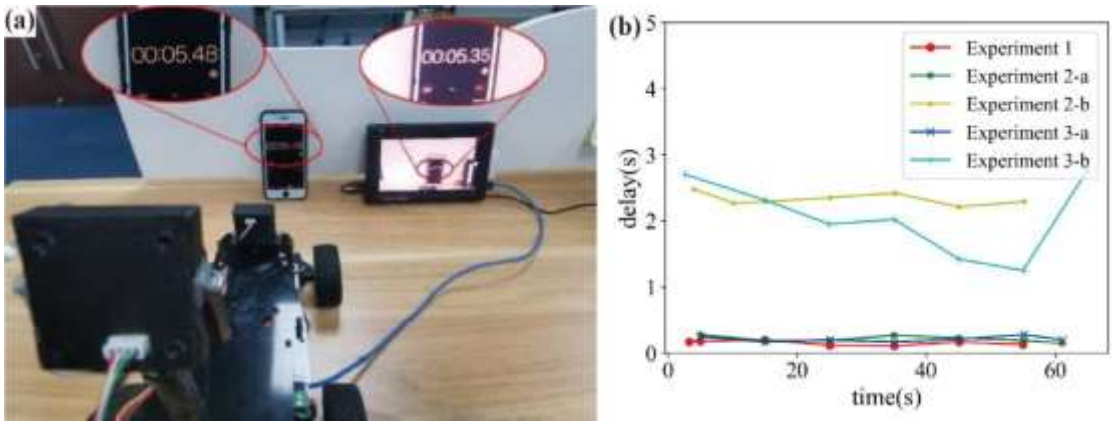


Fig 16: Method diagram of camera testing

As displayed in Fig 16(a), the actual time is obtained from the car camera, which shoots the stopwatch in real-time, and the test time is recorded from the experiments at 10s intervals. The final measurement results are depicted in Fig 16(b). it can be seen from the result of Experiment 1 and Experiment 2-a that after the image transferring by the Redis database, the delay of the data just slightly increases with an average delay within 0.3s, this mainly can be traced back to the time consumption caused by the conversion between pictures and base64 encoding. While a sharp delay can be founded in Experiment 2-b and 3-b, which is mainly due to the uncontrollability of the network. Obviously, this phenomenon is unacceptable for real-time control applications, and it also shows that "cloud-based autonomous driving" is highly dependent on the network. In summary, the experiments fully verified that the Redis-based framework used in our visual navigation car has superior performance in real-time.

4.3 Road Segmentation and Independent Line Patrol Testing

The destination of the road segmentation test is to evaluate the accuracy of the visual navigation car in judging the road conditions. Different positions have been adopted to place the car, and then the corresponding recognition results have been recorded and summarized in Fig 17. Furthermore, the result of road segmentation can be converted into the deviation value of the car from the center of the road. According to that value, the car can adjust its position to ensure that it is always driving in the middle of the road. The experiments also demonstrate whether the straight road or the roads with a large turning radius, the car can well keep driving along the middle of the road.

*Forest Chemicals Review*
*www.forestchemicalsreview.com*
*ISSN: 1520-0191*
*May-June 2022 Page No. 23-44*
*Article History: Received: 24 February 2022, Revised: 05 April 2022, Accepted: 08 May 2022, Publication: 30 June 2022*


Fig 17: (a) real (b) recognition segmentation and (c) AR photo of the camera

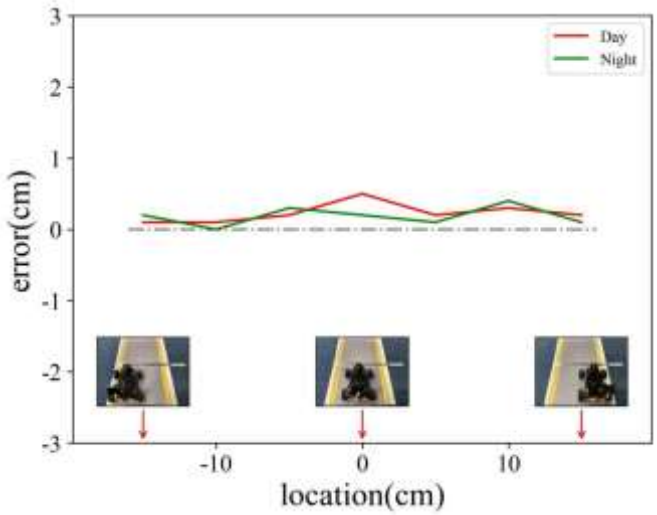


Fig 18: Road segmentation results for different light environments

In addition, in order to verify the robustness of the recognition program, we perform the same test as above on the car under different lighting environments. The results are recorded and summarized in Fig 18. It suggests that the car is running stable in different lighting environments with an error within 0.5cm, that is, no significant impact has been found in the road segmentation recognition result from the light. This desired result can be explained by the coordinated work of edge detection, HSV color detection, and straight-line detection used in dealing with road segmentation, which can minimize the interference of light conditions.

4.4 Traffic Signs Identification

The achievement of traffic sign recognition is by RFB target detection and SSD classification algorithms. Several representative signs are chosen here for recognition testing, as shown in Fig 19(a). Before officially identifying, The model is trained through a self-made data set, and then the trained model is deployed on EAIDK310. The real-time detection results are displayed in Fig 19(b) and (c). Attributing to the Tengine inference framework existing in EAIDK310, the results reflect that the designed car system can achieve accurate recognition with a frame rate of 10-15fps, which is sufficient for the real-time requirements.

*Forest Chemicals Review*
*www.forestchemicalsreview.com*
*ISSN: 1520-0191*
*May-June 2022 Page No. 23-44*
*Article History: Received: 24 February 2022, Revised: 05 April 2022, Accepted: 08 May 2022, Publication: 30 June 2022*


Fig 19: The diagram of (a) traffic signs, (b)-(c) traffic sign identification result

## V. CONCLUSION

This paper constructed a complete platform for visual navigation algorithm testing. A low-cost embedded EAIDK310 development platform with an AI acceleration framework has been chosen as the main processor. Meanwhile, the novel dual-camera scheme is adopted for data acquisition, which can save time for CPU segmentation and transformation of images and apply it in a meaningful computing task. For the auxiliary processor, a multi-functional control board based on the STM32 MCU has been designed. It is demonstrated that the designed self-driving car scheme can efficiently satisfy most of the functional requirements for developing and verifying the visual navigation platform. As this paper focuses on the architecture design of the platform, so it is necessary to conduct in-depth research in achieving the certain function and related algorithms.

## ACKNOWLEDGEMENTS

The authors gratefully acknowledge ARM China for supplying the development kit.

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

## REFERENCES

[1] Grigorescu S, Trasnea B, Cocias T, et al. A survey of deep learning techniques for autonomous driving . Journal of Field Robotics, 2020, 37(3): 362-86.
[2] Du Q, Feng F. Design of intelligent escort robot based on Raspberry Pi . Information Technology, 2021, (07): 78-83.
[3] Kong DX, Zhang QH, et al., et al. System Design of Video Capture Intelligent Car Based on STM32 and Raspberry Pie . Instrument Technique and Sensor, (12): 4.
[4] Shirke YS, Gawade US, Hawre MM, et al. Implementation of SDC: Self-Driving Car based on Raspberry Pi. 2020.
[5] Wu BT, Kong JP, et al. design and implementation of intelligent car based on Arduino and Raspberry Pi . Electronic Design Engineering, 2017, 25(15): 58-61.

*Forest Chemicals Review*
*www.forestchemicalsreview.com*
*ISSN: 1520-0191*
*May-June 2022 Page No. 23-44*
*Article History: Received: 24 February 2022, Revised: 05 April 2022, Accepted: 08 May 2022, Publication: 30 June 2022*

[6] Chen BC, Wu W, Guo Y, et al. Automatic Hand-Eye Calibration System of Robot based on ROS . Computer Simulation, 2020, 37(02): 343-8.
[7] Chen D, Yao BY, Optimal Design of Kernal Correlation Filtering Target Tracking Method forMobile Robot Based on ROS . Journal of Computer-Aided Design & Computer Graphics, 2020, 32(12): 1967-75.
[8] Li YQ, Chen CM. Design of mobile robot automatic navigation system based on ROS and laser radar . Modern Electronics Technique, 2020, 43(10): 176-8+83.
[9] Mao XJ. A Systematic Review on Software Engineering for Autonomous Robot . Chinese Journal of Computers, 2021, 44(08): 1661-78.
[10] Testa A, Camisa A, NOTARSTEFANO G. ChoiRbot: A ROS 2 toolbox for cooperative robotics . IEEE Robotics and Automation Letters, 2021, 6(2): 2714-20.
[11] Blass T, Hamann A, Lange R, et al. Automatic Latency Management for ROS 2: Benefits, Challenges, and Open Problems; proceedings of the 2021 IEEE 27th Real-Time and Embedded Technology and Applications Symposium (RTAS), F, 2021 .
[12] Zhi L, Xuesong M. Navigation and control system of mobile robot based on ROS; proceedings of the 2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), F, 2018.
[13] Chaudhry S, Gautam MNB. A Communication Architecture Based on ROS2 for the Control in Collaborative and Intelligent Automation Systems . Psychology and Education Journal, 2020, 57(9): 710-3.
[14] Dong P, Jiang Z, Burns A, et al. Build real-time communication for hybrid dual-os system . Journal of Systems Architecture, 2020, 107: 101774.
[15] Aufrère R, Gowdy J, Mertz C, et al. Perception for collision avoidance and autonomous driving . Mechatronics, 2003, 13(10): 1149-61.
[16] Yu C, Shi J. Research on the Application of Eye Movement Human-Computer Interaction in Jewelry Design; Proceedings of the International Conference on Human-Computer Interaction, F, 2019 . Springer.
[17] Giordano L. AIoT: a Kendryte K210 proof of concept . 2020.
[18] Cass S. Nvidia makes it easy to embed AI: The Jetson nano packs a lot of machine-learning power into DIY projects-[Hands on] . IEEE Spectrum, 2020, 57(7): 14-6.
[19] Basulto-lantsova A, Padilla-medina JA, Perez-pinal FJ, et al. Performance comparative of OpenCV Template Matching method on Jetson TX2 and Jetson Nano developer kits; proceedings of the 2020 10th Annual Computing and Communication Workshop and Conference (CCWC), F, 2020 . IEEE.
[20] Ruiqin L, Wenan T, Zhenyu C. Design of Face Recognition Access Entrance Guard System with Mask Based on Embedded Development; proceedings of the Journal of Physics: Conference Series, F, 2021 . IOP Publishing.
[21] El-hassan FT. Experimenting with sensors of a low-cost prototype of an autonomous vehicle . IEEE Sensors Journal, 2020, 20(21): 13131-8.
[22] Deng KL, Zhou FG, Cui C, et al. Intelligent tracking trolley based on KEA . Modern Electronics Technique, 2020, 43(16): 13-7.

*Forest Chemicals Review*
*www.forestchemicalsreview.com*
*ISSN: 1520-0191*
*May-June 2022 Page No. 23-44*
*Article History: Received: 24 February 2022, Revised: 05 April 2022, Accepted: 08 May 2022, Publication: 30 June 2022*

[23] Ariyanto M, Haryanto I, Setiawan J D, et al. Real-Time Image Processing Method Using Raspberry Pi for a Car Model; proceedings of the 2019 6th International Conference on Electric Vehicular Technology (ICEVT), F, 2019 .

[24] Fathy M, Ashraf N, Ismail O, et al. design and implementation of self-driving car . Procedia Computer Science, 2020, 175: 165-72.

[25] Liu S, Huang D. Receptive field block net for accurate and fast object detection; proceedings of the European Conference on Computer Vision (ECCV), F, 2018 .

[26] Liu W, Anguelov D, Erhan D, et al. Ssd: Single shot multibox detector; proceedings of the European conference on computer vision, F, 2016 . Springer.

## Appendix A

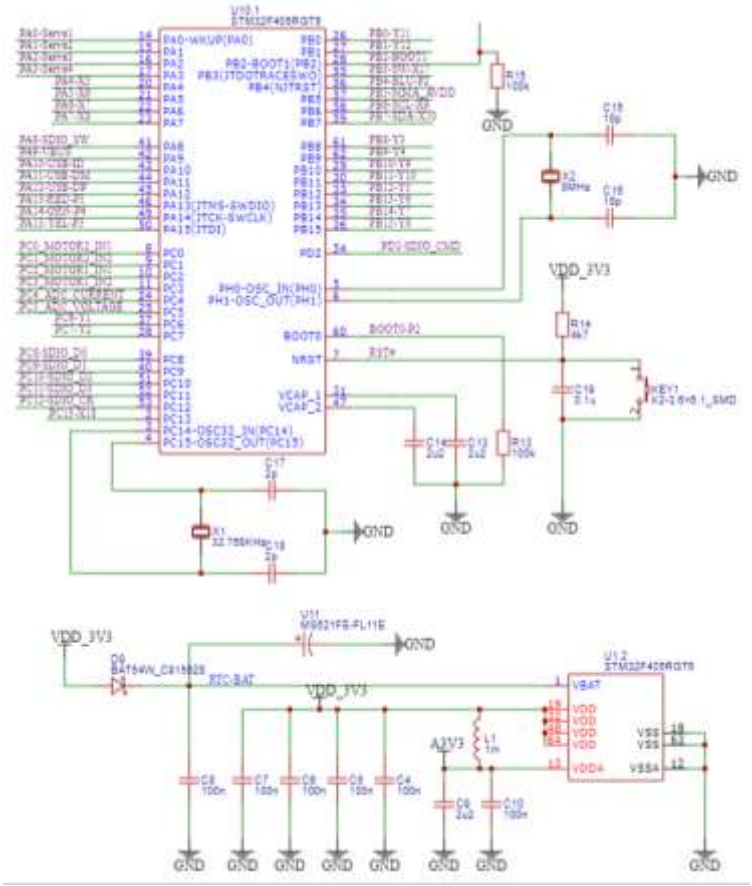


**Fig A.1** Circuit principal diagram of the minimum system

*Forest Chemicals Review*
*www.forestchemicalsreview.com*
*ISSN: 1520-0191*
*May-June 2022 Page No. 23-44*
*Article History: Received: 24 February 2022, Revised: 05 April 2022, Accepted: 08 May 2022, Publication: 30 June 2022*

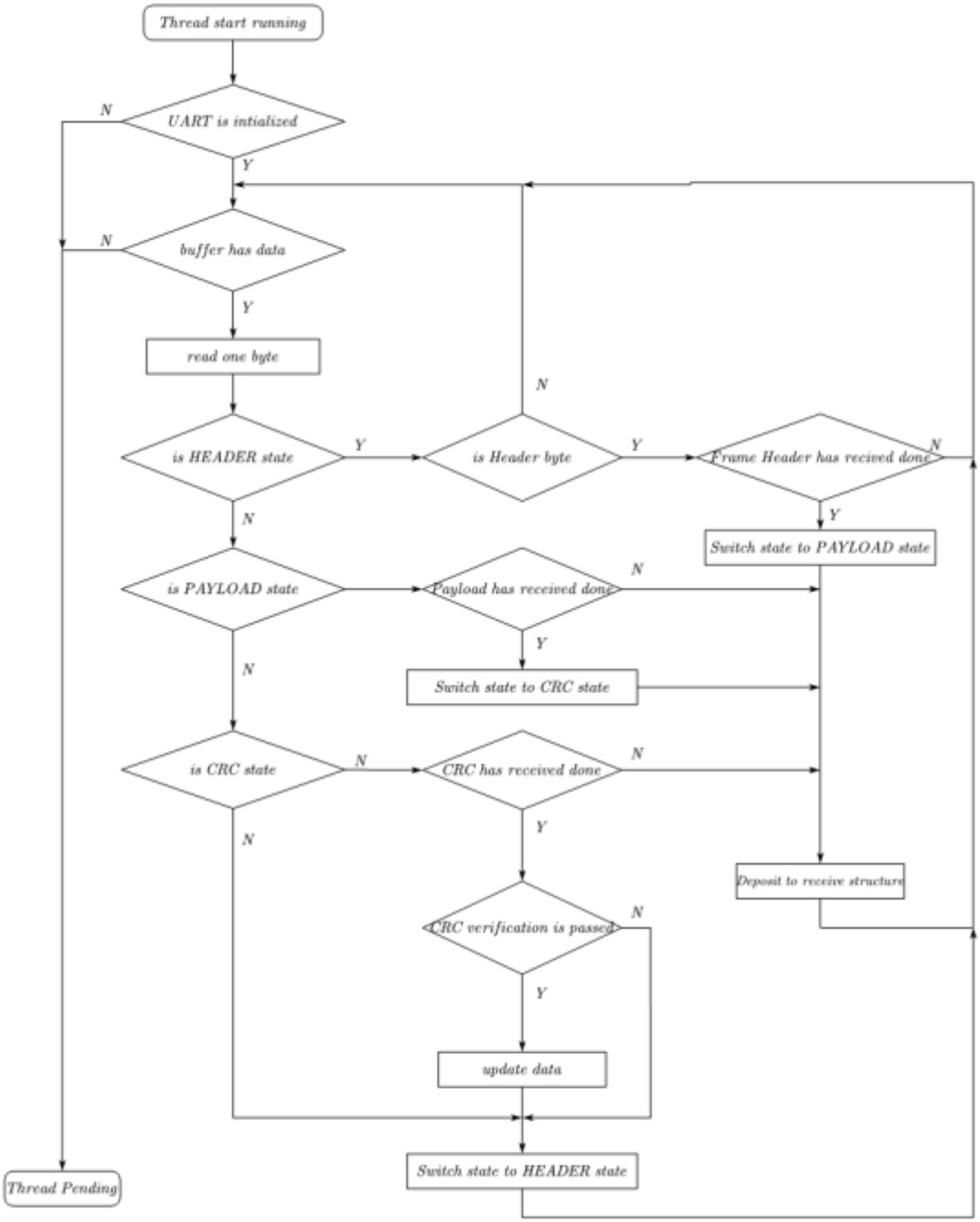


**Fig A.2** Flow chart of data analysis by serial port

*Forest Chemicals Review*
*www.forestchemicalsreview.com*
*ISSN: 1520-0191*
*May-June 2022 Page No. 23-44*
*Article History: Received: 24 February 2022, Revised: 05 April 2022, Accepted: 08 May 2022, Publication: 30 June 2022*

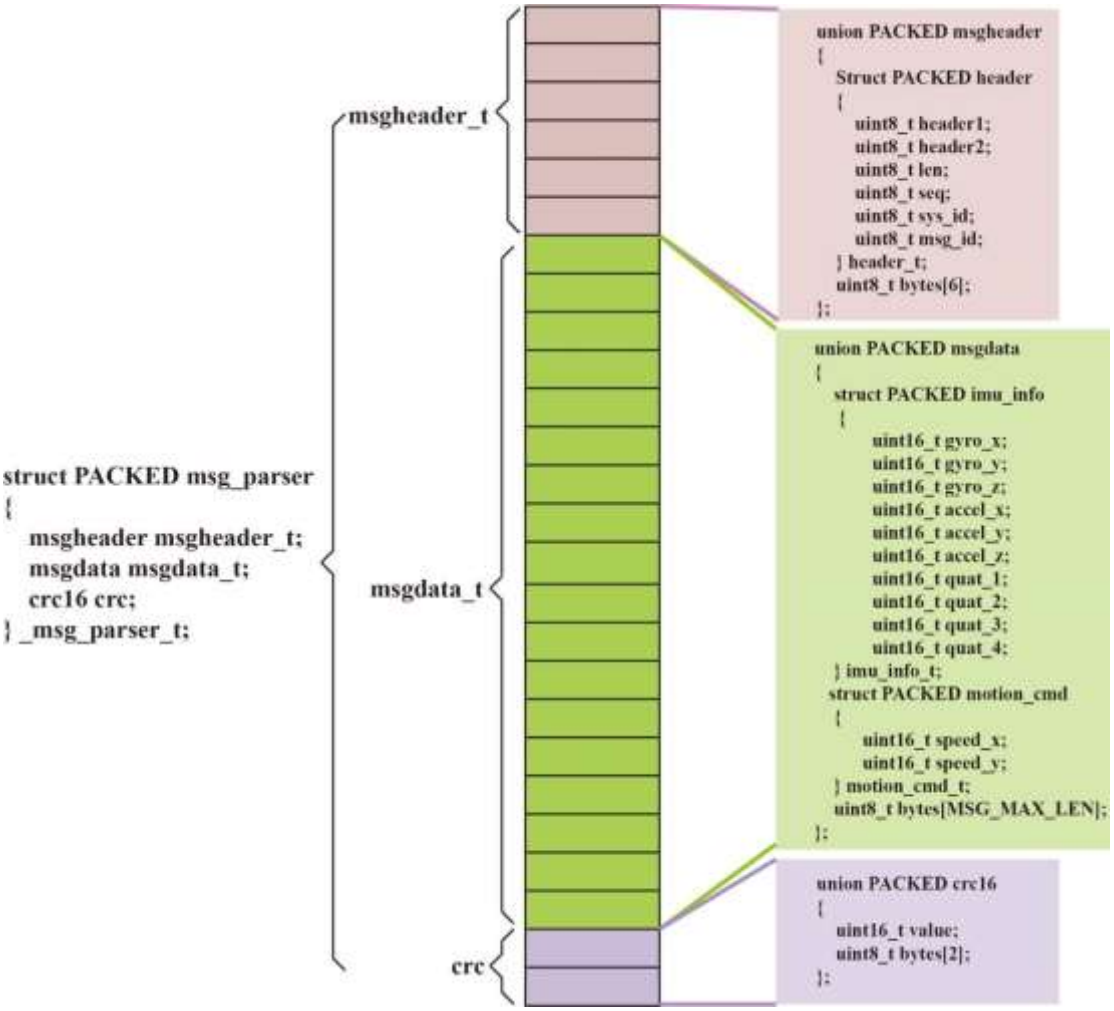


**Fig A.3** The diagram of data structure

**Table A.1** The hardware specifications of EAIDK310

| Parameters | Specification | Parameters | Specification |
|---|---|---|---|
| Soc | RK3228H | WIFI | 802.11ac/a/b/g/n,2.4G/5GHz |
| GPU | ARM4 core 64-bit processor quad-core Cortex-A53, Max: 1.3GHz | Bluetooth | Bluetooth 5.0 |
| GPU | ARM Mali-450 MP2 GPU supporting OpenGL ES 1.1/2.0, OpenVG 1.1 | USB | 1xUSB3,3xUSB2, 1xMicro-USB |

*Forest Chemicals Review*
*www.forestchemicalsreview.com*
*ISSN: 1520-0191*
*May-June 2022 Page No. 23-44*
*Article History: Received: 24 February 2022, Revised: 05 April 2022, Accepted: 08 May 2022, Publication: 30 June 2022*

| | | | |
|---|---|---|---|
| Run memory | LPDDR3 1GB | HDMI | 2.0,1xType-A, Max: 4Kx2K@60Hz |
| Built-in memory | 8GB High-Speed emmc | Debug interface | UART (TTL level) |
| Expand memory | MicroSD supports a maximum of 128GB | Other interfaces | 2xI2C1xSPI9xGPIO2xGPIO (only output) 1xSpeaker |
| Cable network | RJ45,10/100M, adaptive | Power supply | Micro USB 5V / 2A |

**Table A.2** The software specifications of EAIDK310

| Parameters | Description |
|---|---|
| Operating system | Fedora 28, Kernel 4.4, Android8.1 |
| Embedded deep learning framework | Supporting the direct deployment of training frameworks such as Caffe/TensorFlow/Pytorch/MxNet/ONNX/Darknet;<br>Supporting quantified optimization strategies of network performance and layer integration;<br>Providing Unified API (C/Python/JNI) interface and extended interface for the custom operator. |
| High-performance heterogeneous computing library HCL | HCL.NN accelerates the neural network inference operation on the embedded platform;<br>HCL.Vision with common image processing, computer vision, operators and algorithms for mode recognition; and providing heterogeneous scheduling hardware to accelerate chip for image processing;<br>HCL.Audio with common audio signal operators before and after processing, and supporting signal processing methods such as FFT/IFFT, MFCC. |
| Video encoding and decoding of the API | Hard decoding: H264/H265 4K@30fps/60fps;<br>Hard encoding: H264 1080p@30fps. |