

Intelligent garbage sorting and handling truck

Jiaqi Cui

School of Electrical and Control Engineering of North China University of Technology, China

C2324360198@163.com

Abstract

The intelligent garbage sorting truck sends the motor drive signal through the stm 32 main control board, the Raspberry Pi is equipped with ros operating system for autonomous movement, and the yolov5 algorithm, real-time analysis and detection of garbage and classification.openmv The camera identifies the target area of garbage delivery. Arm grasp linkage, through the steering gear control board to drive the steering gear rotation at a certain Angle to grab the ground garbage, and put it on the center panel. The double panel includes a rotary turntable and is accurately driven by a 360° rudder to place the sorted waste in the designated area detected by the openmv.

Keywords

Yolov5 , Stm 32 , Openmv , PID cascade control, Deep learning, Traffic Line detection, Serial communication, Wifi.

1. Introduction

By combining science and technology, mechatronics, circuit logic design, Internet of Things, deep learning algorithms and other practical problems are solved. The wheat wheel chassis is driven to run through PID motor control, combined with ros operating system and 3D real-time map building navigation, autonomous movement, combined with the operation mode of mechanical arm, equipped with steering gear control board, to realize all-round and multi-angle garbage picking. Raspberry PI is equipped with deep learning algorithm to sort garbage in real time, conduct lane detection through openmv Hof transform, and drive the platform to rotate through 360° rotating steering gear to place garbage in designated areas.

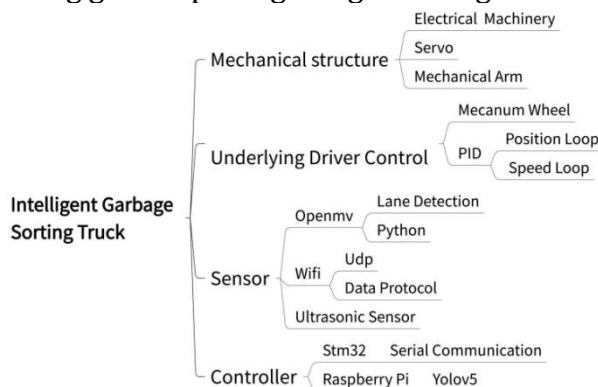


Fig. 1 Overall design process

2. Physical construction

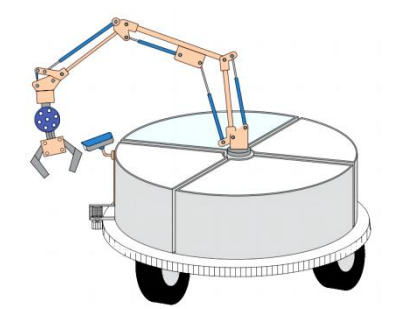


Fig. 2 mechanistic model

Table 1 Mechanical parameters

Mechanical size	Parameter
Steering machine torsion	15kg
Total body weight	20kg
Maximum climbing Angle	$\alpha = 45^\circ$
General travel speed	For 1 m/s
Tire diameter	d=5cm
coefficient of rolling friction	$\mu = 0.24$
transmission efficiency	$\eta = 0.92$

2.1. Motor type selection



Fig. 3 DC brushless motor

Using DC brushless motor, the advantage of DC motor is that it can easily adjust the speed by changing the voltage, and can provide a large torque. Rated power 0.5W-10W, rated speed 100rpm, with incremental encoder, can be used for speed control feedback in DC motor. In such motors, an armature or rotor with a wound wire rotates in the magnetic field produced by the stator. The DC motor encoder provides a mechanism for measuring the rotor speed and provides closed-loop feedback for the drive to achieve precise speed control. Change the armature voltage, the boost speed up, the step down.

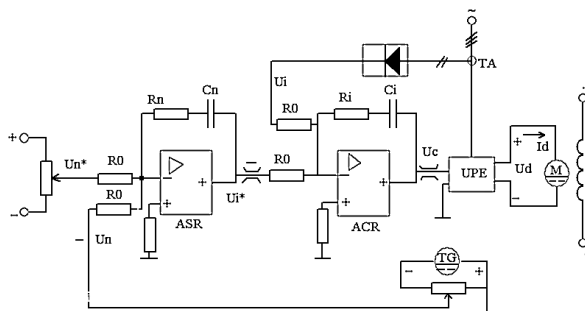


Fig. 4 Schematic diagram of the motor speed regulation

2.2. The underlying control mode

Using McNam wheel, the car can complete all-round movement, compared to the general operation mode, is more stable, flexible, easy to control, simple working place. Turn the two wheat wheels on the same side of the vehicle in the opposite direction, so that the forces along the front and rear sides are offset, and the lateral forces are just in the same direction. The McNamon wheel is an omni-directional wheel that can move in all directions. Because the application scenario is a daily life, more simple and efficient models are needed.

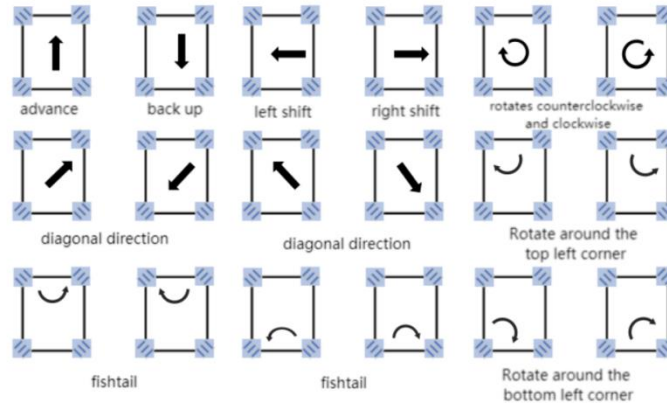


Fig. 5 Schematic diagram of McNam wheel

3. The underlying control

3.1. Encoder speed measurement

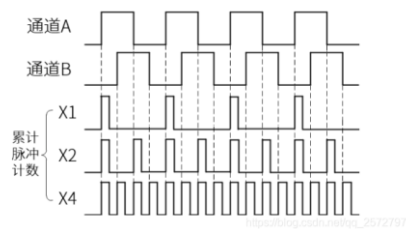


Fig. 6 Working schematic diagram of the encoder

Record the number of lines of the encoder is the number of pulses required for the motor to rotate once, and the deceleration ratio is the number of rotating circles of the motor which corresponds to the number of rotating circles on the output shaft. Read A phase and phase B pulse according to the working principle of encoder and calculate the speed with M measurement $n = \frac{M}{CT}$.

```

2 #include "Encoder.h"
3 #include "stm32f1xx_it.h"
4
5 int actual_speed_A;
6 int actual_speed_B;
7 int actual_speed_C;
8 int actual_speed_D;
9
10 void BianMaqi_A(void)//A
11 {
12     actual_speed_A=(((float)countA-32765)/13/ REDUCTION_RATIO)*100.0f*60.0f;
13     countA=32765;
14 }
15
16 void BianMaqi_B(void)//B
17 {
18     // actual_speed_B=(((float)countB-32765)/ ENCODER_TOTAL_RESOLUTION/ REDUCTION_RATIO)*100.0f*60.0f;
19     actual_speed_B=(((float)countB-32765)/ 13/ REDUCTION_RATIO)*100.0f*60.0f;
20     countB=32765;
21 }
22 void BianMaqi_C(void)//C
23 {
24     actual_speed_C=(((float)countC-32765)/ 13/ REDUCTION_RATIO)*100.0f*60.0f;
25     countC=32765;
26 }
27 void BianMaqi_D(void)//D
28 {
29     actual_speed_D=(((float)countD-32765)/ 13/ REDUCTION_RATIO)*100.0f*60.0f;
30     countD=32765;
31 }
32

```

Fig. 7 Encoder speed measurement code

3.2. PID position loop, speed loop and reference modulation simulation

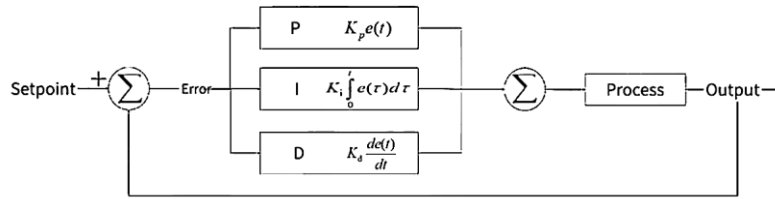


Fig. 7 PID schematic diagram

The PID control principle is combined with the feedback control system, through proportional

control $p_{out} = K_p e(t)$, integral control $I_{out} = K_i \int_0^t e(\tau) d\tau$, differential control for historical deviation $u(t) = K_p [e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{d(e(t))}{dt}]$, current deviation and recent deviation are eliminated respectively.

3.2.1. Position ring

The purpose of adding the position ring is to ensure that the car can move the specified distance, rotate the fixed angle, and can be achieved by changing the number of rotating circles of the motor. According to the discrete PID formula:

$$PWM = Kp * e(k) + Ki \sum e(k) + Kd[e(k) - e(k - 1)]$$

The $e(k)$ is the deviation, the $\sum e(k)$ is the sum of the deviation, and the $e(k - 1)$ is the last deviation.

```

32 int position_pi(int Encoder, int Target)
33 {
34     float Kp=15, Ki=0.01;
35     static int Bias, PWM, Intergral_bias;
36     Bias=Encoder-Target;
37     Intergral_bias+=Bias;
38     PWM=Kp*Bias+Ki*Intergral_bias;
39     return PWM;
40 }
    
```

Fig. 8 Location ring code

3.2.2. Speed ring

Using incremental PID control method:

$$\Delta u [n] = K_p \{e [n] - e [n - 1]\} + K_i e [n] + K_d \{e [n] - 2e [n - 1] + e [n - 2]\}$$

The determination of the control increment $\Delta u (k)$ is only related to the sampling value of the last three times, and it is easy to obtain better control effect by weighting processing; the computer only output the control increment each time, that is, the change of the actuator position, so the influence range when the machine failure is small and will not seriously affect the production process.

3.3. Total control algorithm of wheat wheel chassis

The speed loop position is double loop control, set a target position value, read the current motor position value, bid to calculate the read position data, set the calculation result to the bid value of the motor speed target value, and configure PWM according to the calculation results to control the motor.

```

87 /*
88 int PID_Pos_Cal(PID *Pos_PID, PID *Speed_PID, int Current_Pos, int Current_Speed, int Target)
89 {
90     PID_General_Cal(Speed_PID, Current_Speed, PID_General_Cal(Pos_PID, Current_Pos, Target));
91     return Speed_PID->output;
92 }
    
```

4. Selection of the control panel and the sensor type

4.1. Steering gear control board

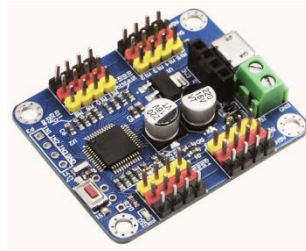


Fig. 9 Steering control board

Connect the steering gear to the steering gear control board and add action groups to the steering gear through the software. Porter rate: 9600bps; the data protocol is as follows:

Table 2 Set the speed of the rudder machine

Data protocol header	instruct	channel	lower byte	high byte
0xFF	0x01	CH	DataL	DataH

Table 3 Set the steering gear position

Data protocol header	instruct	channel	lower byte	high byte
0xFF	0x02	CH	DataL	DataH

Table 4 Set action groups

Data protocol header	instruct	channel	lower byte	high byte
0xFF	0x09	CH	DataL	DataH

4.2. Stm 32 master panel

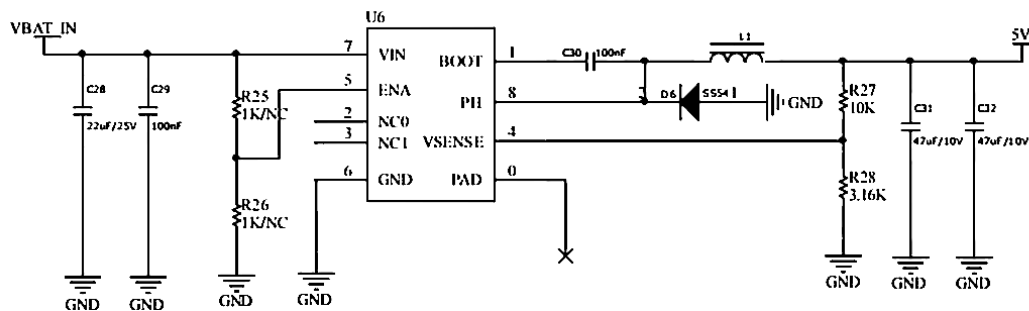


Fig. 10 Circuit diagram of antihypertensive module

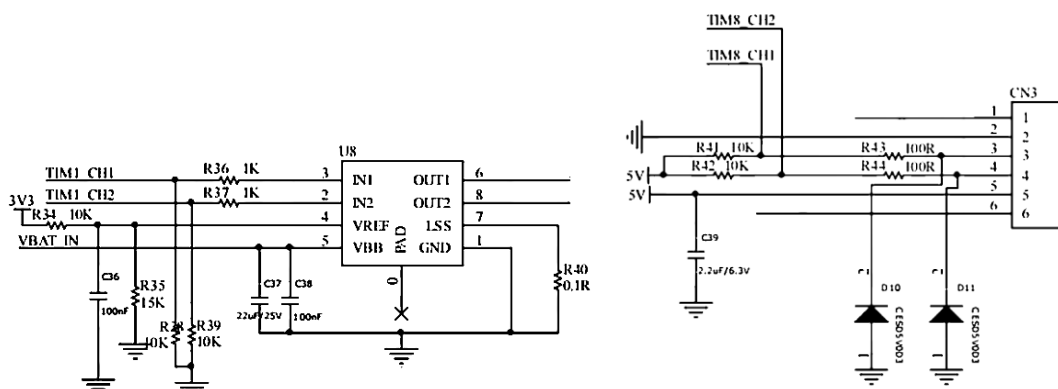


Fig. 11 Circuit diagram of the motor drive module

4.3. Raspberry Pi and Sensors

After the Raspberry Pi and stm 32 are connected, they can make serial port communication to pass on the garbage classification results detected by the Raspberry Pi to stm 32. The ultrasonic module feeds back the obstacle information in real time, and the car avoids the obstacles in time.

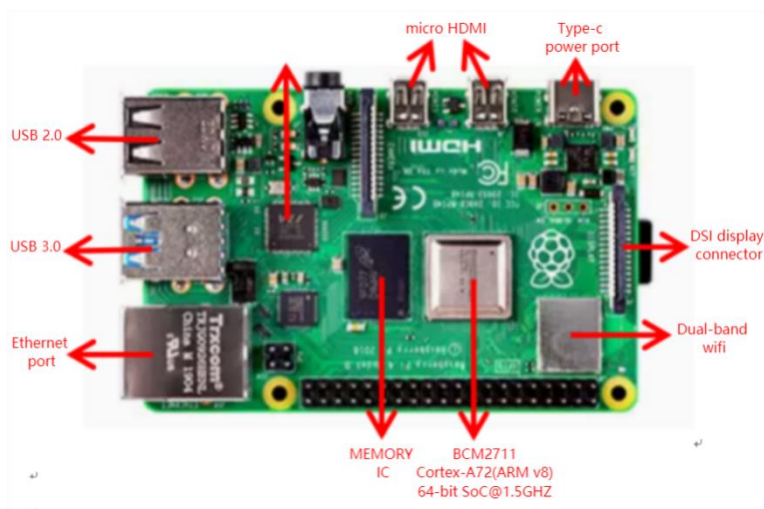


Fig. 12 Raspberry Pie 4B



Fig. 13 The Raspberry Pi is powered by deep learning garbage detection



Fig. 14 Ultrasonic sensor

5. Image recognition

5.1. Openmv sorting area identification

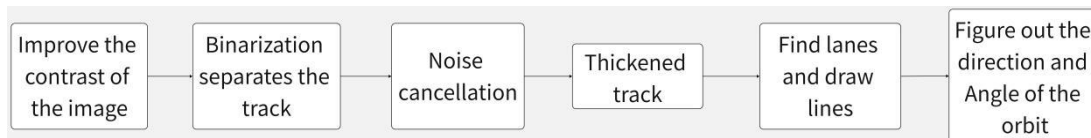


Fig. 15 Openmv Line detection logic diagram

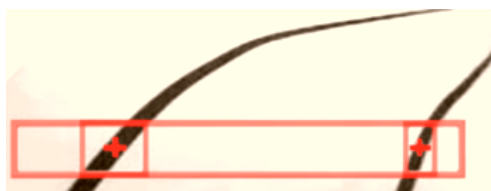


Fig. 16 Openmv Line detection effect

5.2. Yolov5 Garbage classification

Through the training method of Yolov5 deep learning, all kinds of garbage are treated and analyzed, including kitchen waste, other garbage, recyclable garbage and hazardous garbage. Yolov5 The algorithm can be run on the Raspberry Pi to analyze the target items in real time. Depth _ multiple: control the number of submodules, width _ multiple: control the number of convolution cores.anchor The size is preset for the resolution of the input image 640640, which can detect large targets on either small feature graph or small targets on large feature graph.

```
# Parameters
nc: 6 # number of classes
depth_multiple: 0.33 # model depth multiple
width_multiple: 0.50 # layer channel multiple
anchors:
- [10,13, 16,30, 33,23] # P3/8
- [30,61, 62,45, 59,119] # P4/16
- [116,90, 156,198, 373,326] # P5/32
```

Fig. 17 Yolov5s.Yaml parameter configuration

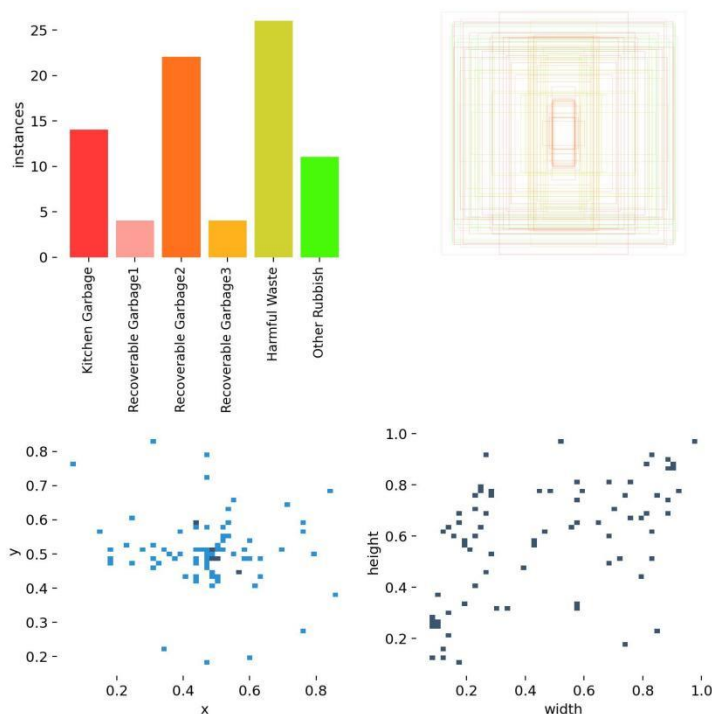


Fig. 18 Yolov5 Training of the model parameters

6. Serial communication

6.1. Stm 32 and the Raspberry Pi communication protocol

According to the communication protocol, stm 32 and Raspberry Pi transmit data to determine the frame head and needle tail and check bit.

```

13 void Camera_Read(uint8_t FunCode, uint8_t *pdata)
14 {
15     uint8_t Data[4];
16     Data[0]=0x5A, Data[1]=FunCode, Data[2]=0x00, Data[3]=0xED;
17     HAL_UART_Transmit(&huart5, Data, sizeof(Data), 1000);
18     HAL_UART_Receive_IT(&huart5, pdata, 5);
19 }
    
```

Fig. 19 Stm 32 terminal Raspberry Pi receiving function

```

if data == 0x5a:
    length = self.Circleloop.get_front_second()
    if length > 1 :
        if self.Circleloop.get_front_second() <= self.Circleloop.get_queue_length():
            databuf = []
            for i in range(length):
                databuf.append(self.Circleloop.get_front())
                self.Circleloop.dequeue()

            if (databuf[length-1]) == self.crc_byte(databuf, length-1):
                pass
            else:
                # print databuf
                # print "Crc check Err %d"%self.crc_byte(databuf, length-1)
                pass

            #parse receive data
            if(databuf[3] == 0x04):
                self.Vx = databuf[4]*256
                self.Vx += databuf[5]
                self.motor_pwm = databuf[6]*256
                self.motor_pwm += databuf[7]
                self.steering_angle_back = databuf[8]*256
                self.steering_angle_back += databuf[9]
    
```

Fig. 20 Raspberry Pi side receiving function

7.2. Check the garbage delivery and classification

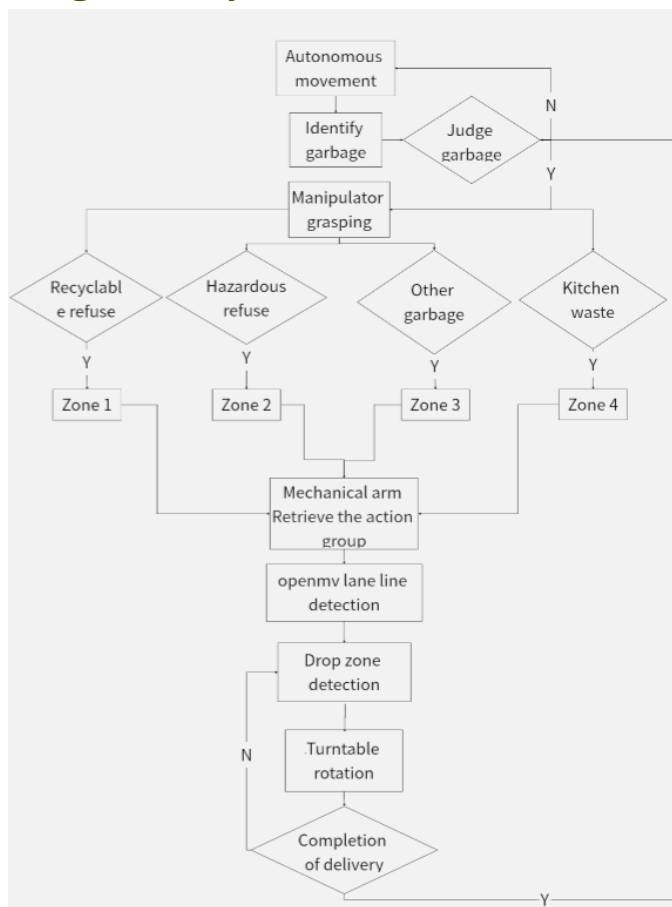


Fig. 24 Logical diagram of garbage classification and delivery

8. Conclusion

Through the intelligent garbage sorting truck, it independently navigates and moves in the community, and monitors the garbage on the ground in real time through the camera. According to the image recognition technology, the garbage is recycled and sorted to the designated destination, and the 360° rotating steering gear rotates and puts the garbage at the designated classification point. Through this technology, reduce the cleanliness of the community, improve efficiency, and make the community life more healthy and hygienic.

References

[1] An Sa, Lian Xiaoqin, Cheng Kaiyuan, et al. Simulation system of unmanned intelligent Vehicle based on OpenMV [J]. Information Technology and Informatization, 2019,(6).DOI:10.3969/j.issn.1672-9528.2019.06.002.

[2] Y Fu 2021].Phys.Conf.Ser.2083 042036 DOI 10.1088/1742-6596/2083/4/042036 Application of Machine Vision Recognition System in Mobile Robot.

[3] F Tian, Y Xu, H Z Du, X H Guo, X Li, J P Li,2022,DOI: 10.1109/ITAIC54216.2022.9836627,IEEE Joint International Information Technology and Artificial Intelligence Conference (ITAIC).