Part I – Smart Car Racing

The steering angle of the smart car was defined as large, medium, and small in order to realize the tracking function in the tracking car competition. I called the six infrared sensors on the head of the cart (three left and three right) to detect the position of the cart in relation to the black trajectory line to determine which steering angle is used by the cart.

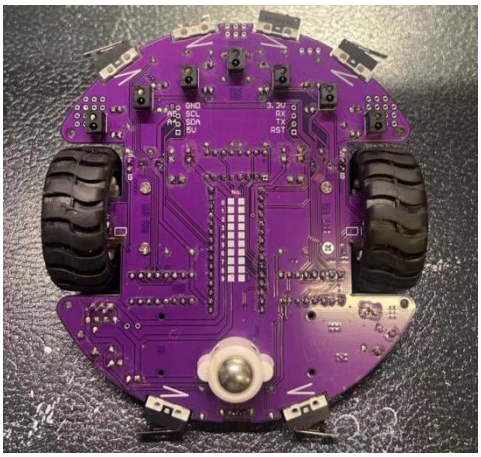

Figure 1. The circuit of the smart car

In the code section in order to implement the sensor data reading of the cart, I called the comm.h function, when the sensor detects a black line, the sensor will output to the arduino. in order to implement the control of the motor of the cart, I called the motor.h function, which allows me to adjust the rotational speed of the motor using the PWM with a range of 0-255.

test4-ir § | Adafruit_NeoPixel.cpp | Adafruit_NeoPixel.h | comm.cpp | comm.h | motor.cpp | motor.h

```
#include "Adafruit_NeoPixel.h"  // 彩色灯珠驱动
#include "comm.h"               // 传感器数据读取
#include "motor.h"              // 电机控制
```

Specifically, when the outermost sensor detects the black line, it means that the cart deviates from the trajectory to the greatest extent, and currently, the cart adopts the large steering, which is defined as the motors on both sides rotating in different directions at the maximum speed to achieve the function of in-situ steering. When the innermost sensor detects the black line, it indicates that the cart slightly deviates

from the trajectory, at this time only needs to be reduced, indicating that at this time the cart slightly deviates from the path, at this time only needs to reduce the wheel speed on one side to form the wheel speed difference, to achieve the small steering function. Medium grade steering serves as a transition between large steering and small steering, resulting in higher accuracy in steering judgement and a smoother steering process for small vehicles.

```
void loop()
{
    static int motor_right = 0; // 初始化电机速度为0
    static int motor_left = 0;  // 初始化电机速度为0
    reload_shift_reg(); // 刷新传感器数据

    if (sensor.ir_left_3) // 左侧第3个(中间)红外传感器测到黑线或悬空
    {
        motor_right = 255;// 设定大左转
        motor_left = -255;
    }


    else if (sensor.ir_right_3) // 右侧第3个(中间)红外传感器测到黑线或悬空
    {
        motor_right = -255;
        motor_left = 255; // 设定大右转
    }


    else if (sensor.ir_left_1) // 左侧第1个(内测)红外传感器测到黑线或悬空
    {
        motor_right = 255;// 设定小左转加大右电机转速
        motor_left = 200;
    }

    else if (sensor.ir_left_1) // 右侧第1个(内测)红外传感器测到黑线或悬空
    {
        motor_right = 200;
        motor_left = 255; // 设定小右转加大左电机转速
    }
```

```
else if (sensor.ir_left_2) // 左侧第2个(中间)红外传感器测到黑线或悬空
{
    motor_right = 255;// 设定中左转加大右电机转速
    motor_left = 180;
}

else if (sensor.ir_left_2) // 右侧第2个(中间)红外传感器测到黑线或悬空
{
    motor_right = 180;
    motor_left = 255; // 设定中右转加大左电机转速
}

else //全速前进
{
    motor_right =255;
    motor_left = 255;
}
```

In order to increase the running speed of the trolley, we adopt the method of replacing the original four 1.5V dry batteries with 12V lithium batteries, which significantly reduces the mass of the trolley and increases the maximum speed of the motor at the same time. We constantly tweaked and tested the wheel speed differentials and rpm, resulting in a balance between motor speed, sensor response, and Arduino arithmetic. This allows the car to travel at full speed in a straight line, and to turn in time at corners without running off the track.

Warning： During the test we found that cp1 cm3 electrolytic capacitor cm4 ceramic chip capacitor three capacitors very easy to damage.