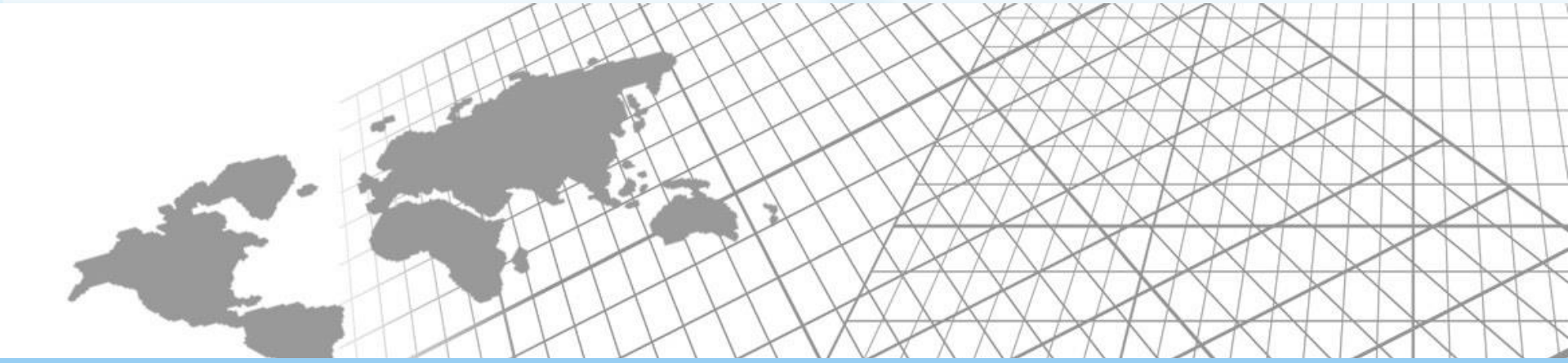
北航 自动化科学与电气工程学院
School of Automation Science And Electrical Engineering



National Intelligent Car Contest for College Students (Freescale Cup)

Benchun Zhou

School of Automation Science and Electrical Engineering
Beihang University(Beijing, China)

CONTENTS

CONTENTS

school of Automation Science And Electrical Engineering

北航自动化科学与电气工程学院

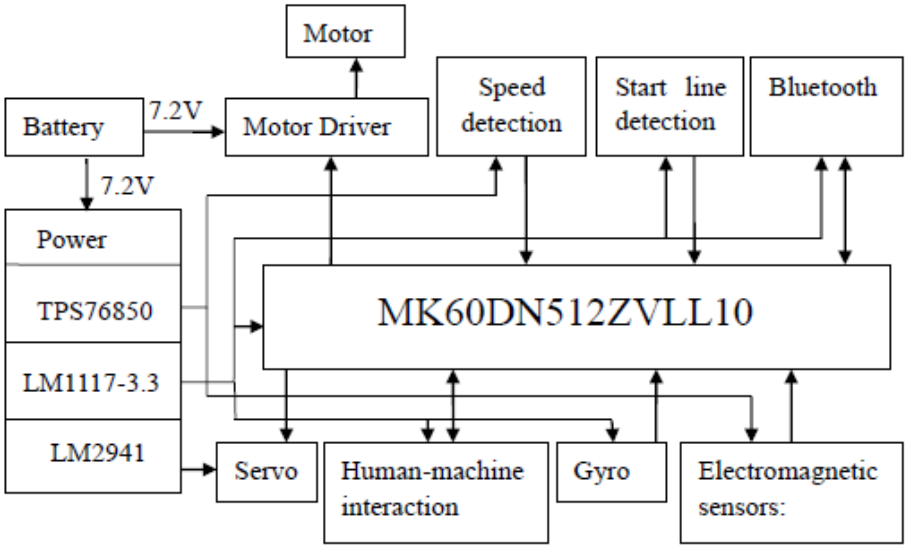
1. Introduction

- **1.1 Competition Introduction**

- In the background of the 10th National Intelligent Car Contest for College Students, this article introduces the software and hardware structures and the development flow of the vehicle control system. This contest adopting A-type car model prescribed by the contest organization committee, using the 32-bit MCU K60 produced by Freescale Semiconductor Company as the core controller, requires the car finish the race in the fastest speed. The whole system includes the aspects of the mechanism structure adjustment, the sensor circuit design and signal process, control algorithm and strategy optimization etc. It captures the road information through resonant circuit, then abstracts the road position. After that, PD feedback control is used on the steering. At the same time, the system obtains the current speed using a speed sensor, so that it can realize the feedback control of the speed by PID method.

1. Introduction

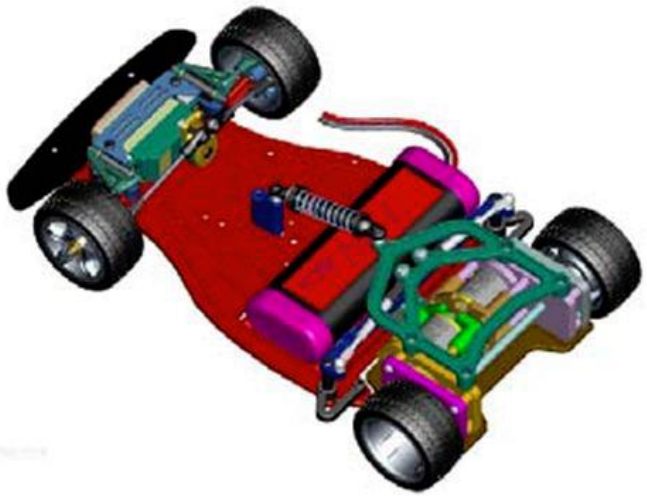
- **1.2 Main Work**
- Completed a four-wheel car engineering production to follow the track with alternating current (20KHz, 10mA).
- Designed control schemes, including electromagnetic sensor circuit, motor driver circuit (hardware) and steering servo control algorithm (software).
- Distributed sensors reasonably, obtained the position of the car and controlled the speed and direction using Segment PID algorithm.



CONTENTS

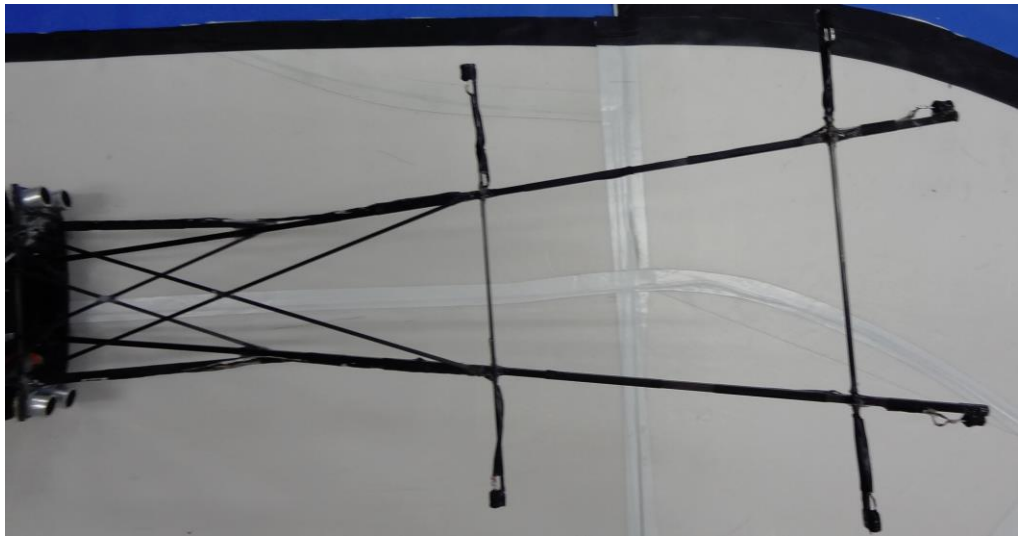
2. Mechanical Engineering

- **2.1 Basic Car Model**
- Car model requirements:
  - Length: no requirement
  - Width: no longer than 250mm
  - Sensor: no more than 16
  - Motors: RN260
  - Power: no higher than 25V
  - Capacity of components : no bigger than 2000uF



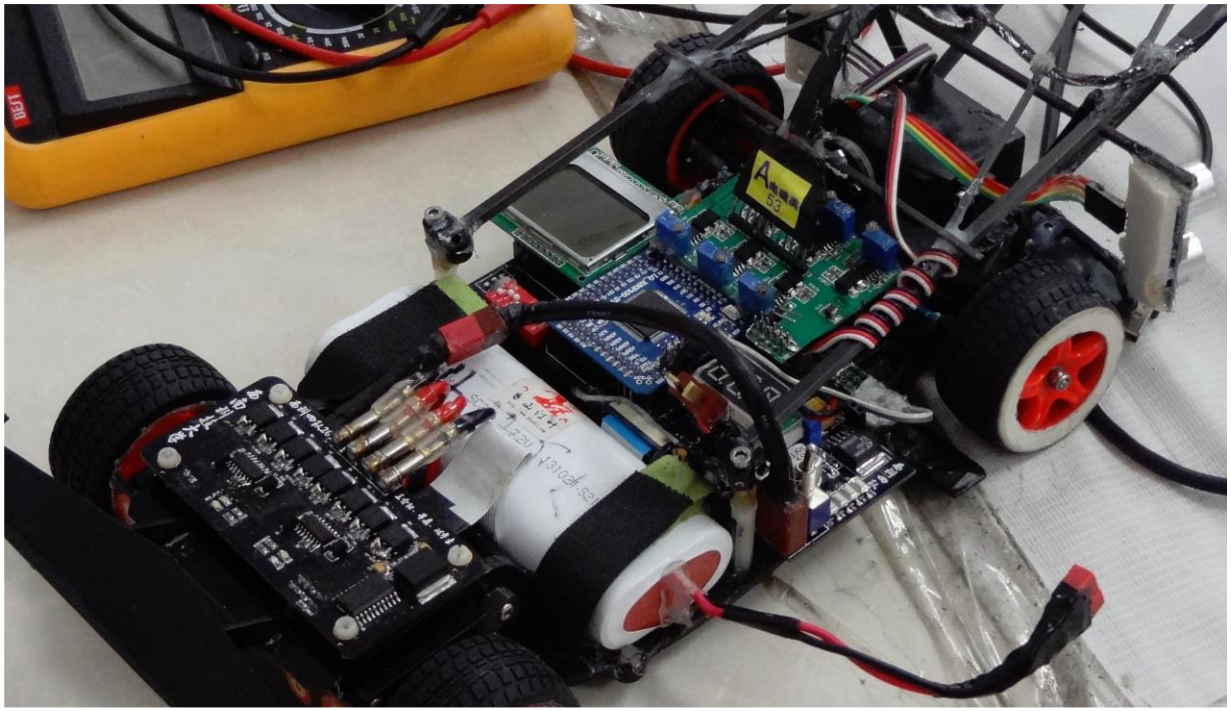
2. Mechanical Engineering

- **2.2 Sensors**
- Electromagnetic sensors:
  - 6 Electromagnetic sensors
  - Distribute in the front of the car
  - Length: 24.9cm
  - Width: 24.9cm



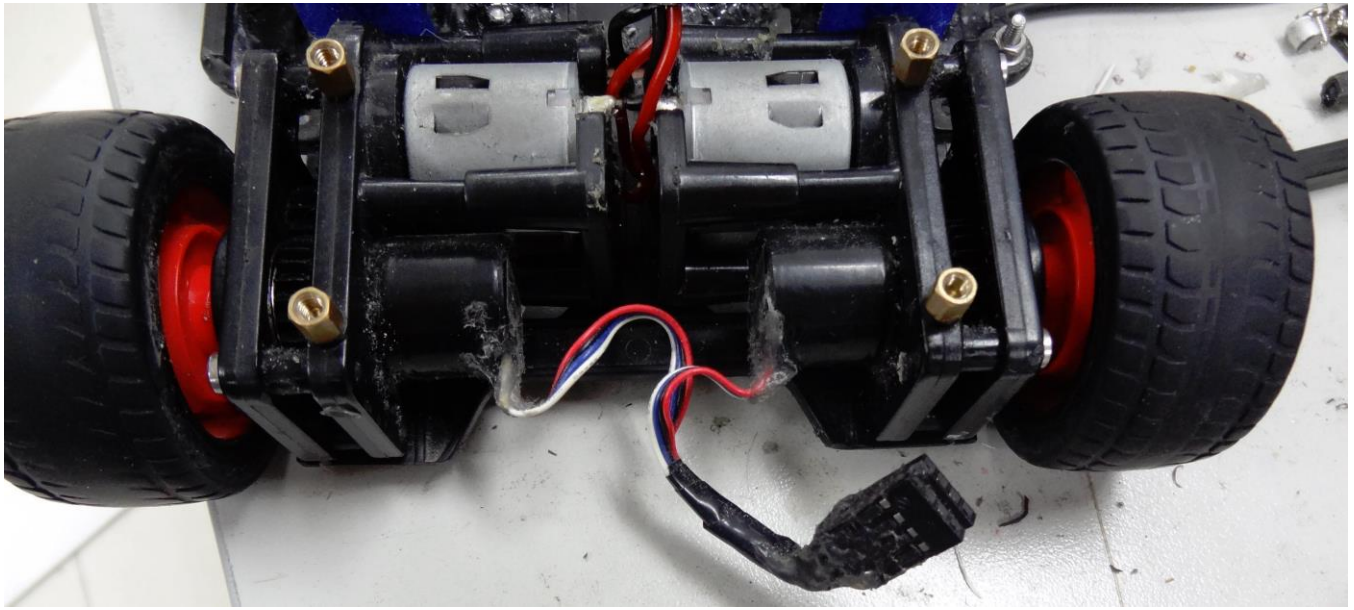
2. Mechanical Engineering

- **1.2 PCB**



2. Mechanical Engineering

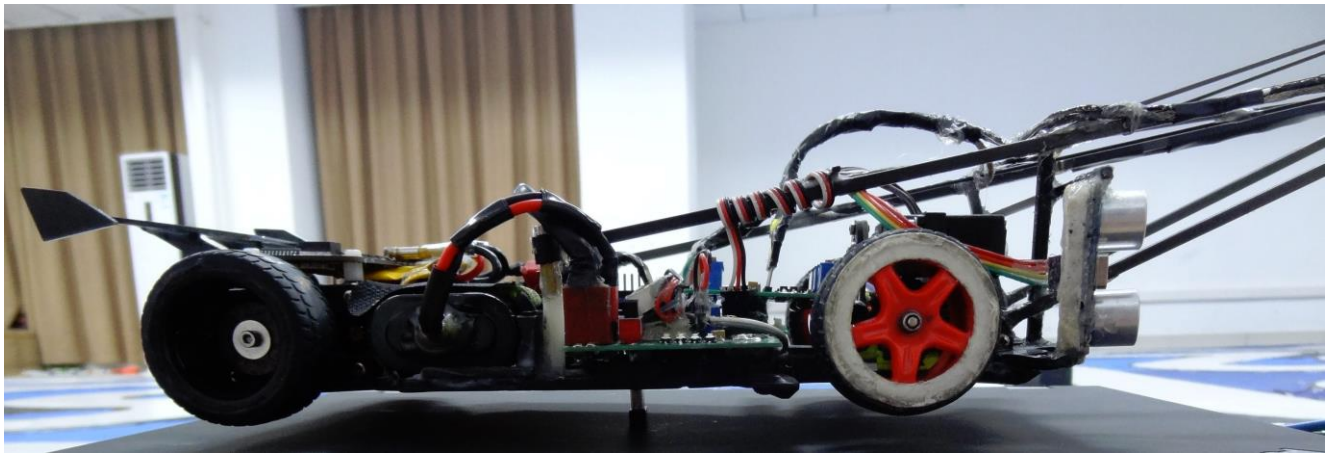
- **2.3 Encoders**
- Photoelectric encoder
  - 256 lines



2. Mechanical Engineering

- **2.4 Gravity Center**

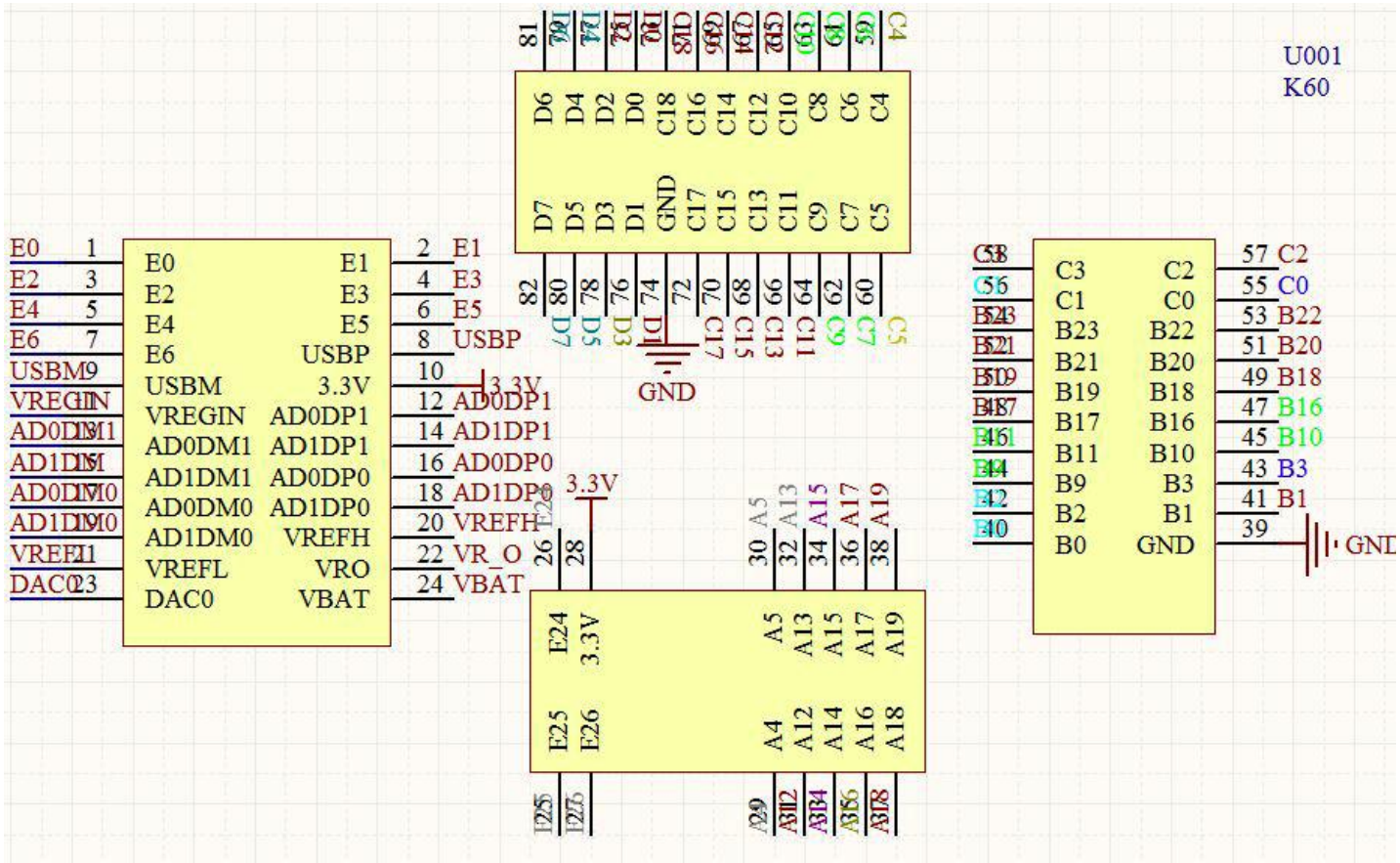
- Servo adjustment
- Front-wheel adjustment
- Adjustment of rear differential gear



CONTENTS

3. Hardware Design

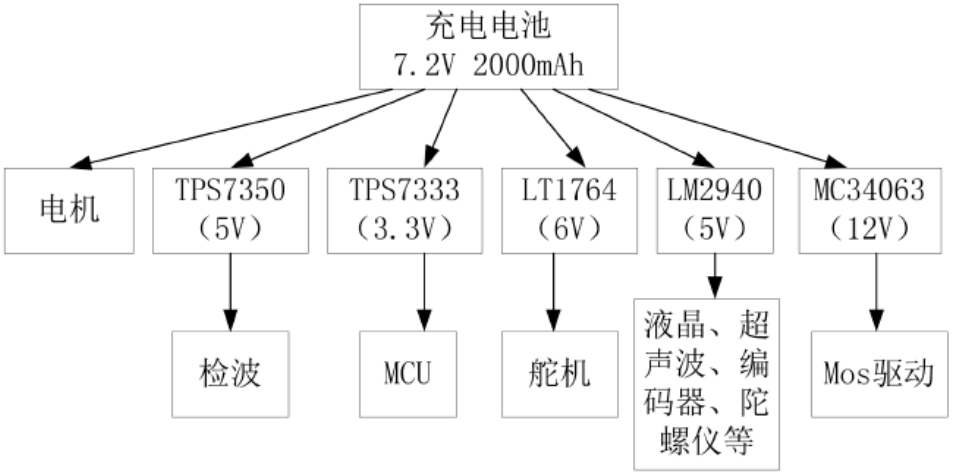
- **3.1 MCU**
- MK60DN512ZVLL10

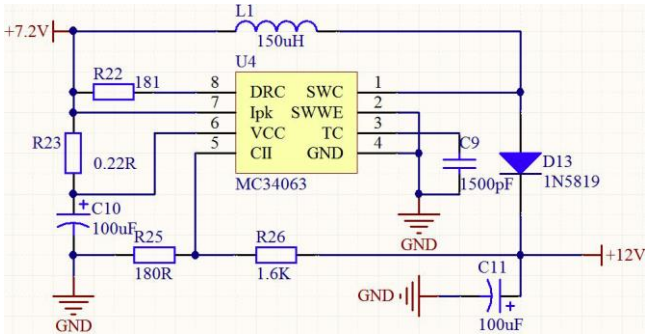


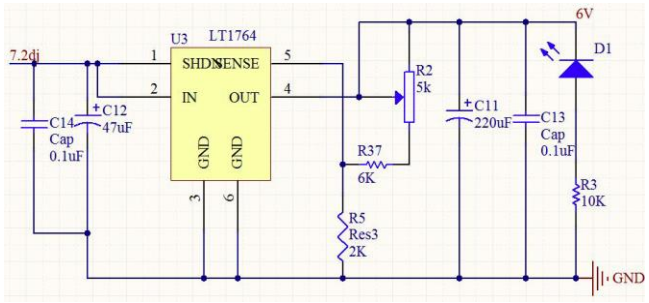
3. Hardware Design

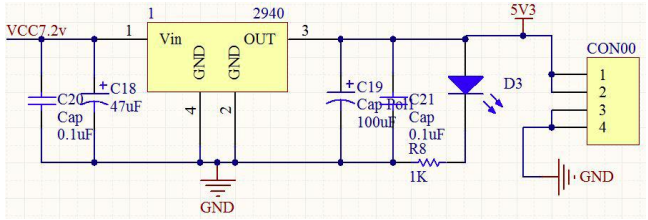
- **3.2 Power Management**

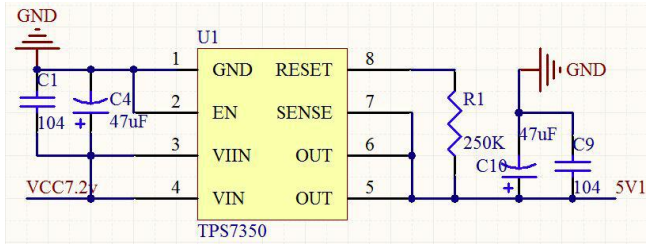
- Battery: 7.2V, 2000mAh
  - Motors: 12V
  - Signal process: 5V
  - MCU: 3.3V
  - Servo: 6V
  - Display: 5V

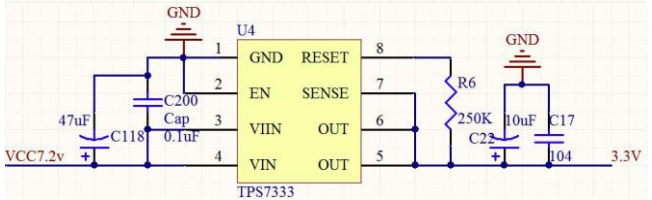








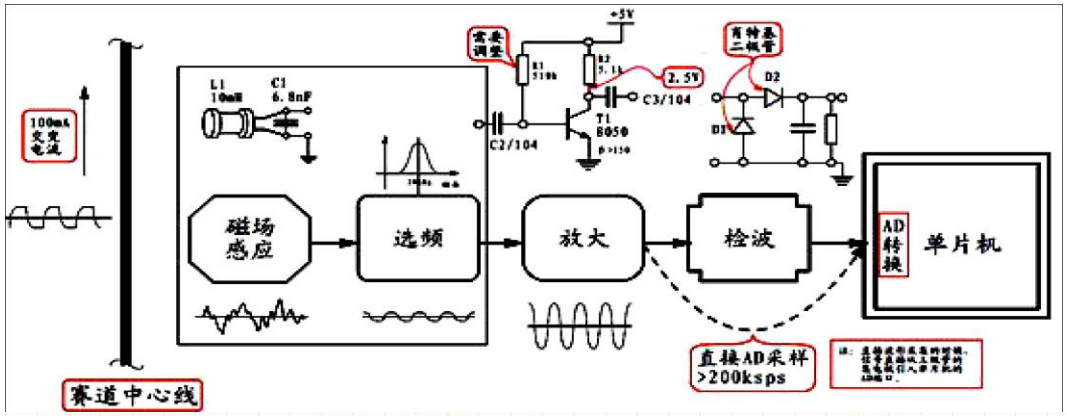


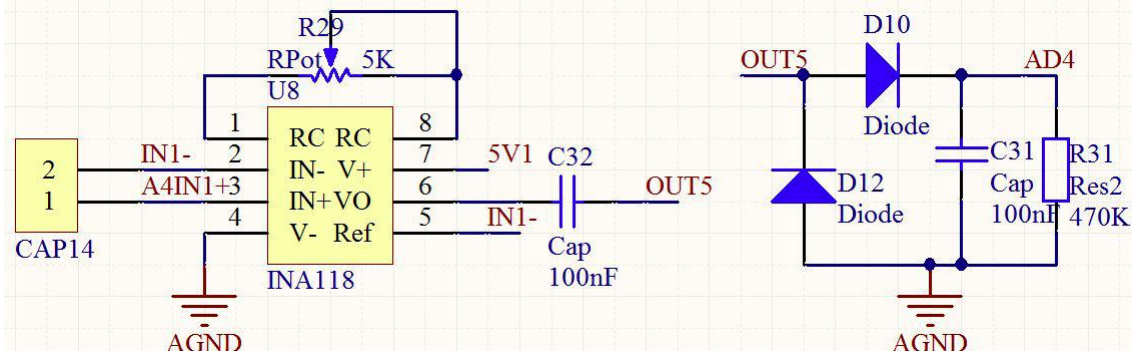


3. Hardware Design

- **3.3 Electromagnetic Sensors**

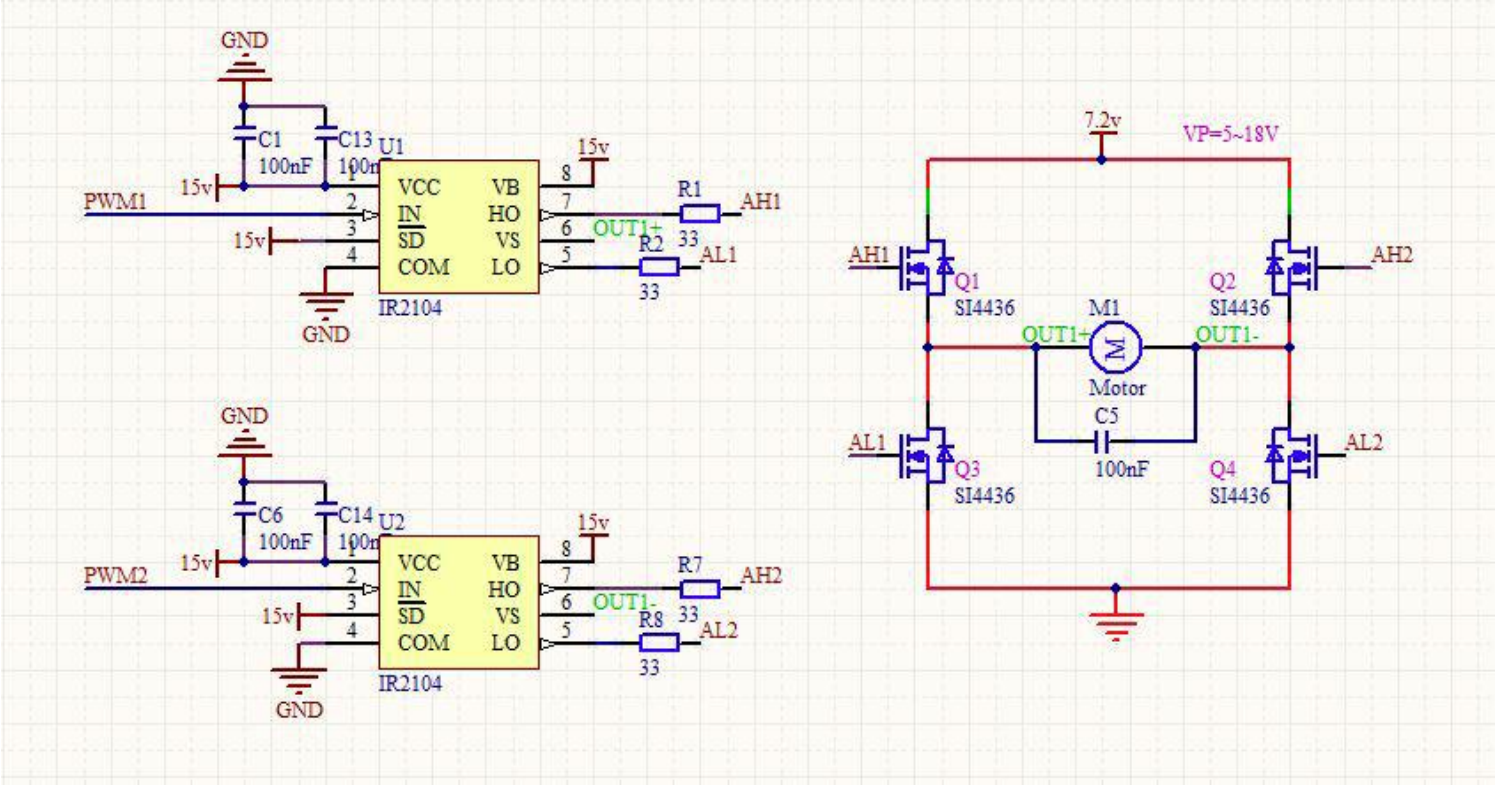
- The electromagnetic sensors are design to detect the path of track. There is an alternating current (20KHz, 10mA) under the path, so we designed signal process circuit with inductance(10mH), capacitor (6.8nF), and amplifier, which can detect the signal from the path and indicate how far between the car and the path





3. Hardware Design

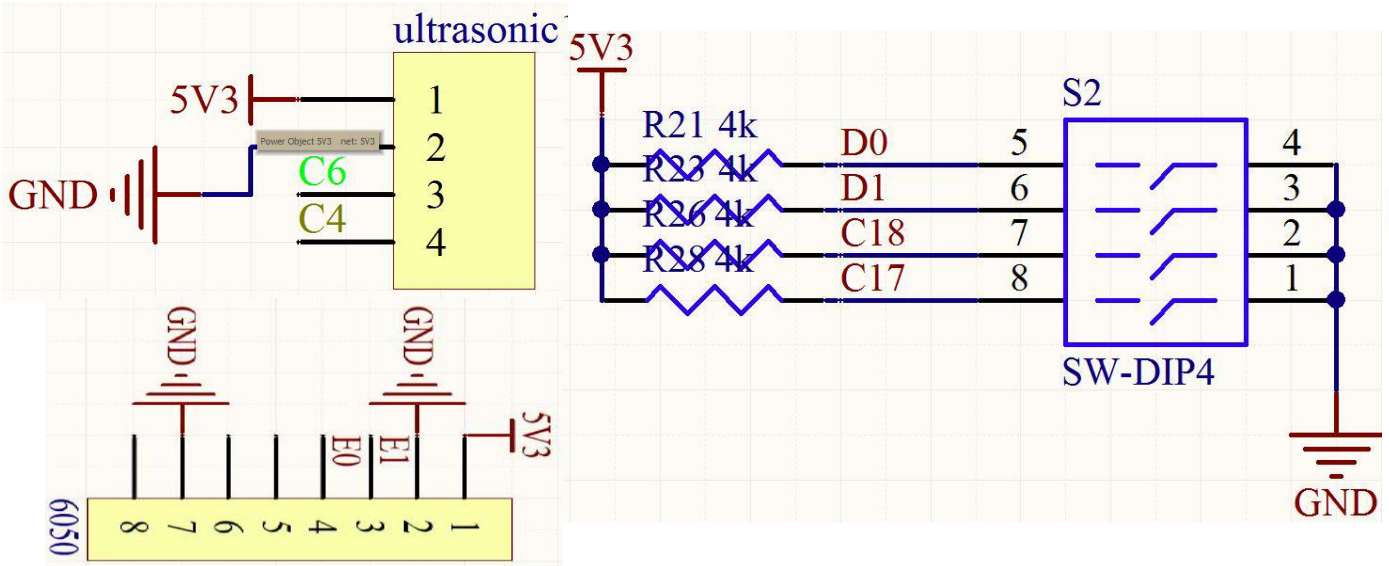
- **3.4 Motors**
- To drive the motors, we build the H-Bridge circuit, which can control the motor to run forward or reverse. NMOS is chosen.



3. Hardware Design

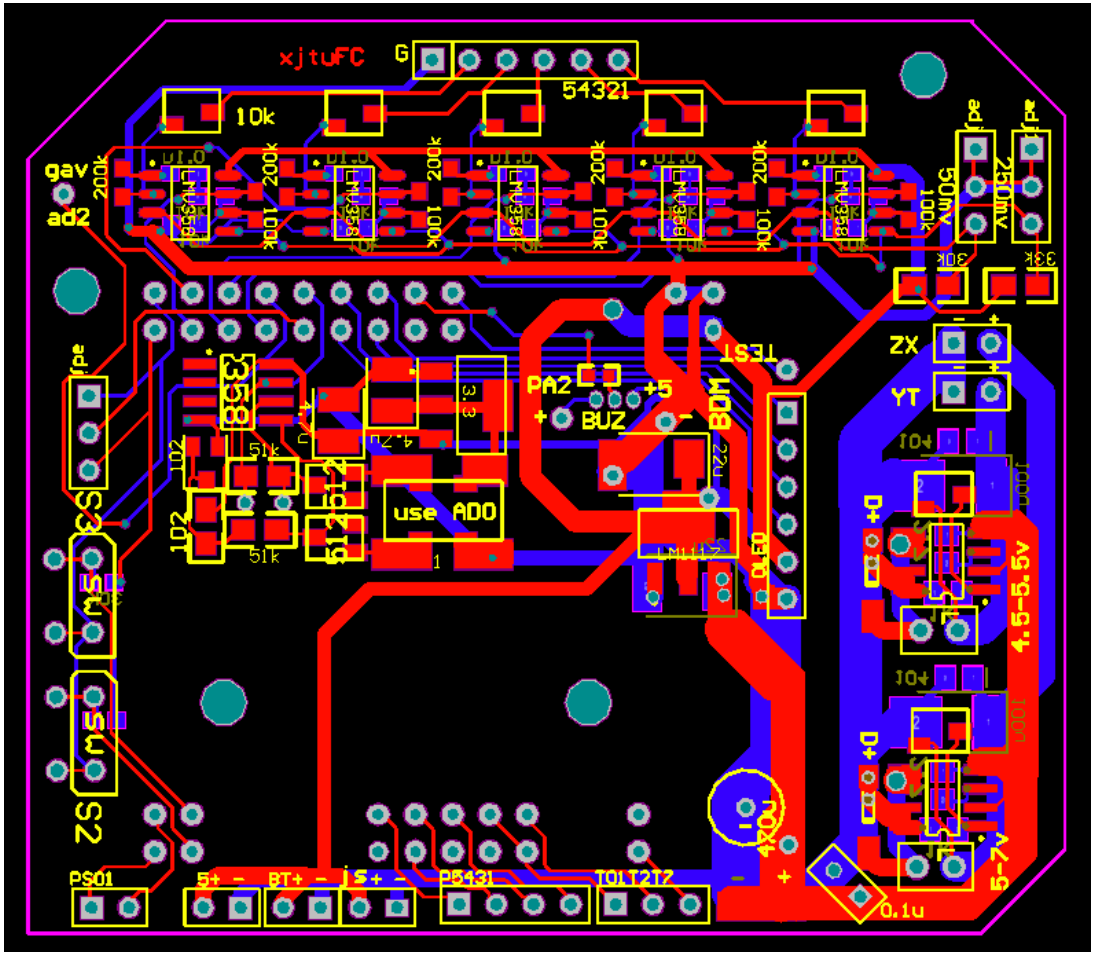
- **3.5 Encoder and others**

- Encoders are design to detect the speed of the car.
- Other sensors such as Ultrasonic, Bluetooth are also used for debug.



3. Hardware Design

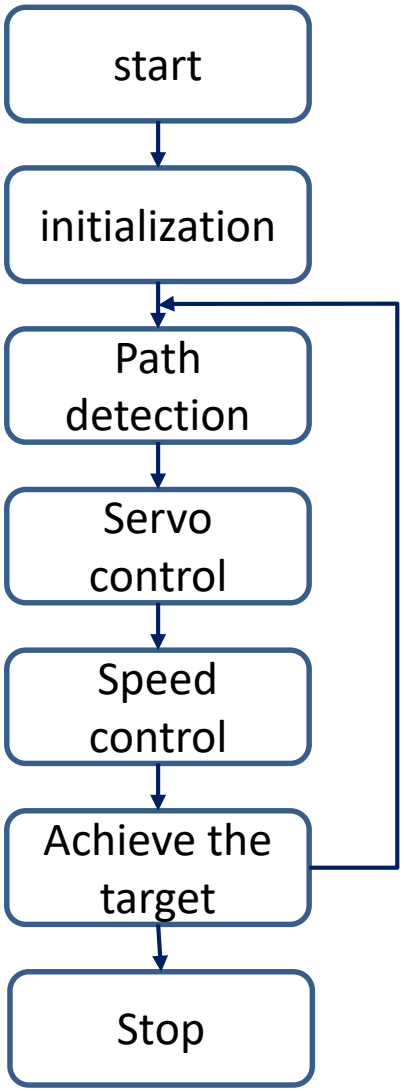
- **3.6 Overall**



CONTENTS

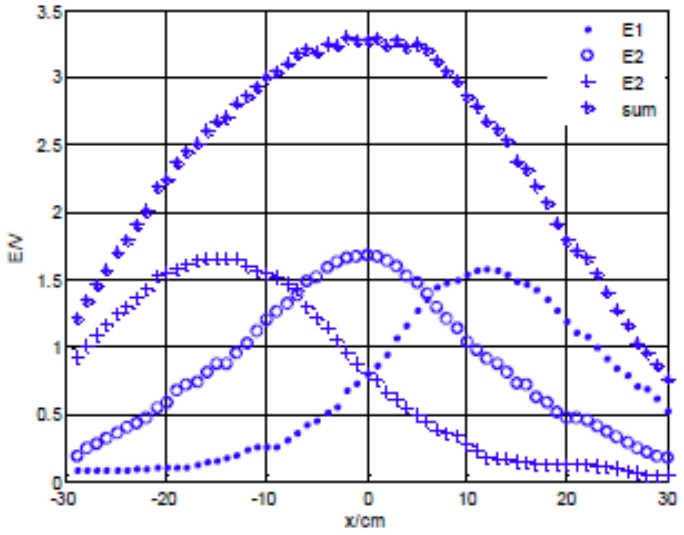
4. Software Design

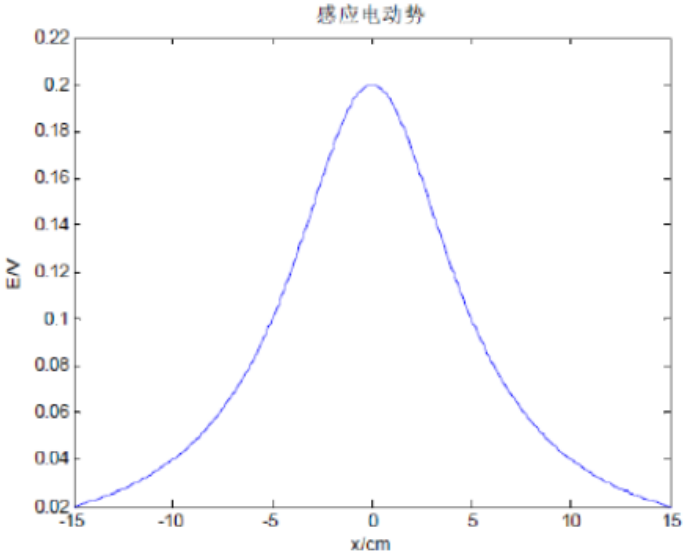
- **4.1 Overall Design**



4. Software Design

- **4.2 Path Detection**
- We have several sensors in different place to detect the path. Data fusion is employed to estimate the position of the car

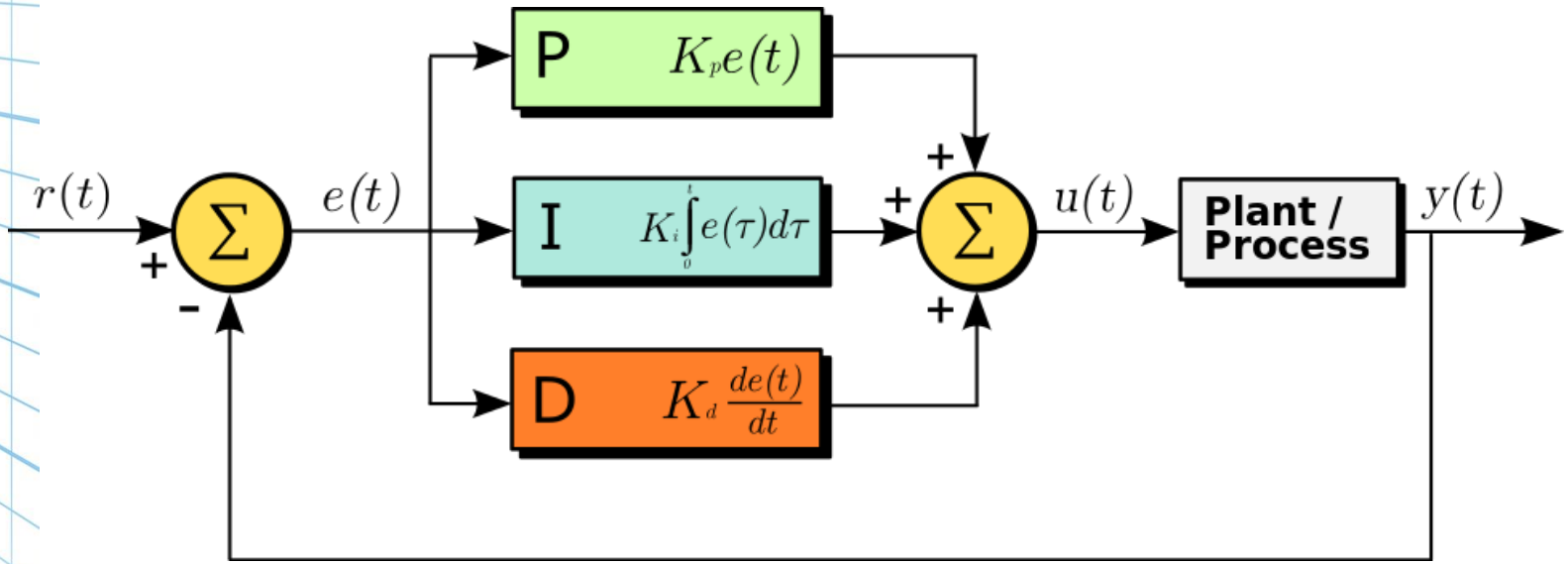




4. Software Design

- **4.3 Servo Control Algorithm**
- PD controller

$$\Delta u(k) = K_p\left[e(k) - e(k-1)\right] + K_d\left[e(k) - 2e(k-1) + e(k-2)\right]$$



4. Software Design

- **4.4 Motor Control Algorithm**
- PID controller

$$\Delta u(k) = K_p\left[e(k) - e(k-1)\right] + K_i e(k) + K_d\left[e(k) - 2e(k-1) + e(k-2)\right]$$

$$\Delta u(k) = q_0 e(k) + q_1 e(k-1) + q_2 e(k-2)$$

$$q_0 = K_p(1 + \frac{T}{T_i} + \frac{T_d}{T}) \qquad q_1 = -K_p(1 + \frac{2T_d}{T}) \qquad q_2 = K_p\frac{T_d}{T}$$

$$u(k) = u(k-1) + \Delta u(k)$$

CONTENTS

5. Competition