
人工智能小车第一阶段



实验报告

李雅娴 2015201940

杨宁宁 2015201942

张宇宁 2015202036

一、实验目的

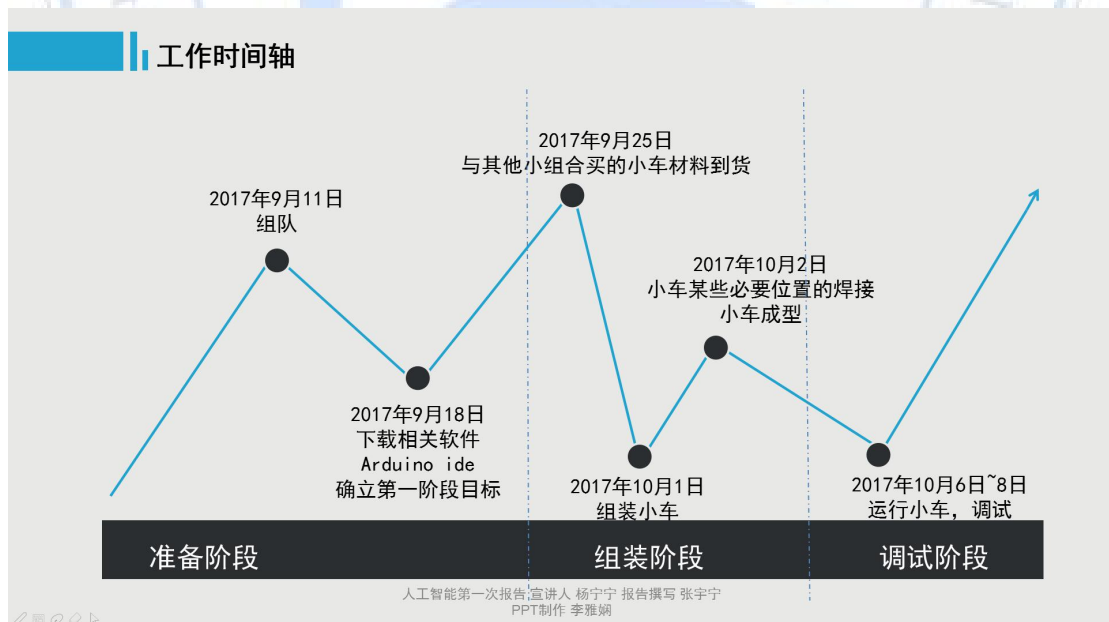
人工智能是指由人制造出来的机器所表现出来的智能，被称为“第四次工业革命”，我们要实现的小车基础功能是：

- 1、能够自动前进，不原地打转；
- 2、遇到障碍物时，能够通过超声波规避障碍物；
- 3、能够使用电脑或者手机蓝牙遥控小车，使其完成前进、左右转弯、后退、停止、启动；
- 4、能够向电脑或者手机传拍摄的图像以及视频；

在实现了小车基础功能后，我们要实现的小车高级功能是：

- 1、通过装人工智能软件，使小车能够沿固定的轨道行驶；
- 2、实现声控人工智能小车

二、实验步骤



1、元件准备

UNO 板*1

蓝牙模块*1

超声波探测器*1

扩展板*1

面包板*1

9V 蓄电池*1

充电宝*1

亚克力板车架*1（配有马达、轮胎）

焊接笔*1

螺丝刀*2

Mp3 扩展板*1

发光二极管若干

杜邦线若干

2、配置软件环境并导入代码

我们在电脑上下载 Arduino id 软件，将 UNO 与电脑相连，将代码导入 UNO 板。





```
sketch_oct01a | Arduino 1.8.5 (Windows Store 1.8.10....  —  □  ×
文件 编辑 项目 工具 帮助

sketch_oct01a

#include <AFMotor.h>

AF_DCMotor Rback_motor(4);
AF_DCMotor Lback_motor(1);
float RLRatio = 1.0;

int maxSpeed = 80; //最大速度
char getstr; //串口数据
int IntervalTime; //测距时间
int trig = 9; //sero1保留端口
int echo = 10; //sero2保留端口
int control=1; //0:人工控制;1:自动控制

void setup() {
    // put your setup code here, to run once:
    Serial.begin(9600);
    Rback_motor.setSpeed(0);
    Rback_motor.run(RELEASE);
}

编译完成。
项目使用了 5870 字节, 占用了 (18%) 程序存储空间。最大为 32256 字节。
全局变量使用了286字节, (13%)的动态内存, 余留1762字节局部变量。最大为2048字节。

14 Arduino/Genuino Uno 在 COM1
```

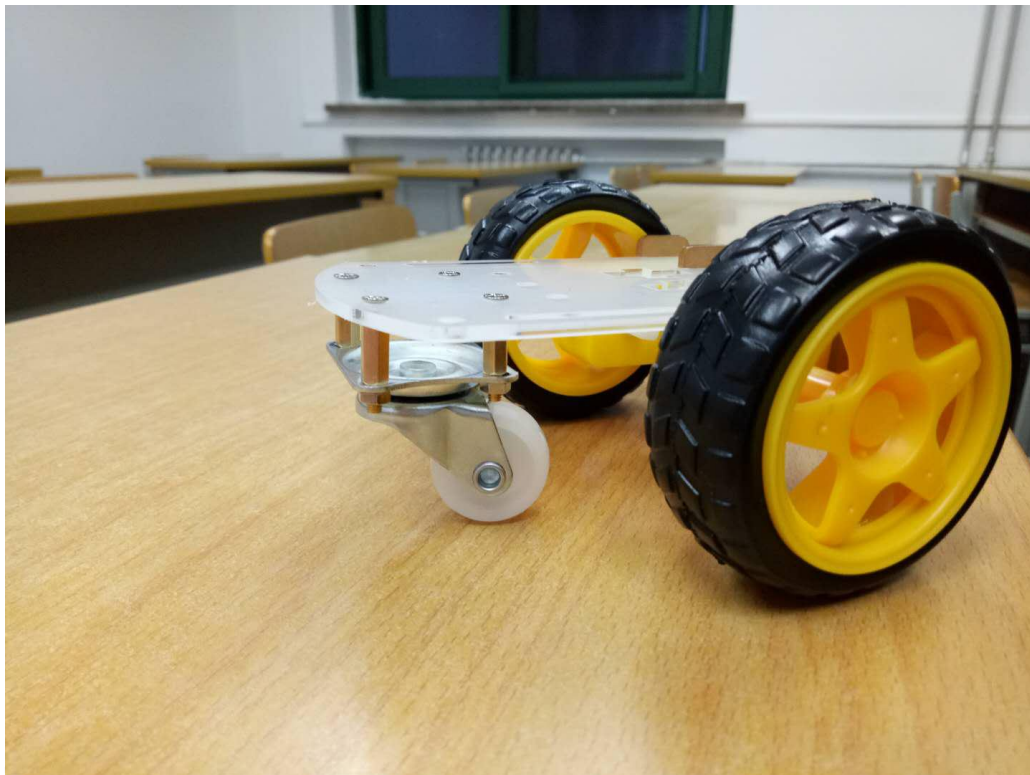
(图说：代码编译导入过程)

3、小车拼装

根据老师在课堂上讲解的内容以及拍摄的照片，我们使用杜邦线将各个部分电路连接起来。具体操作如下：

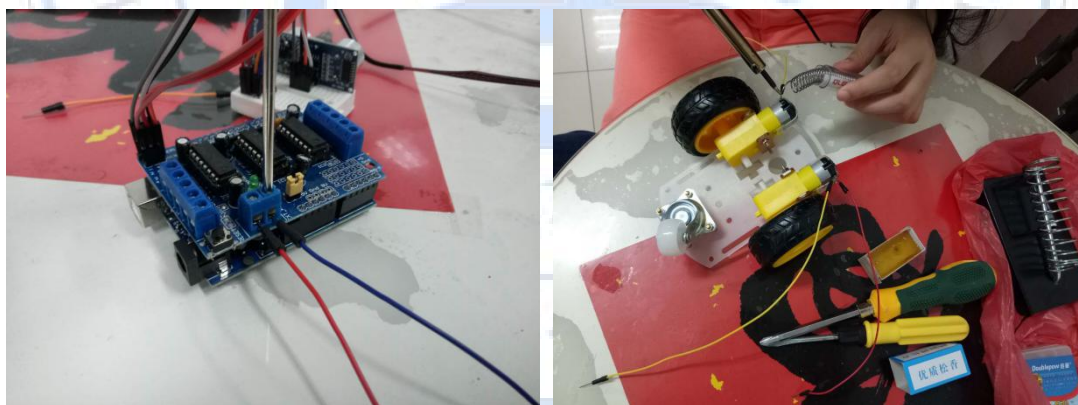
(1) 拼装小车

一共有两个马达，三个轮胎，我们使用螺丝钉将轮胎以及马达固定在了小车底盘。



(2) 安装 UNO 板、扩展板以及面包板

我们将扩展板插在 UNO 板上，使用橡皮筋固定在了车架上的后面。面包板固定在车架上的前面。



(3) 安装蓝牙和传感器

我们将蓝牙和传感器错位插在面包板上。

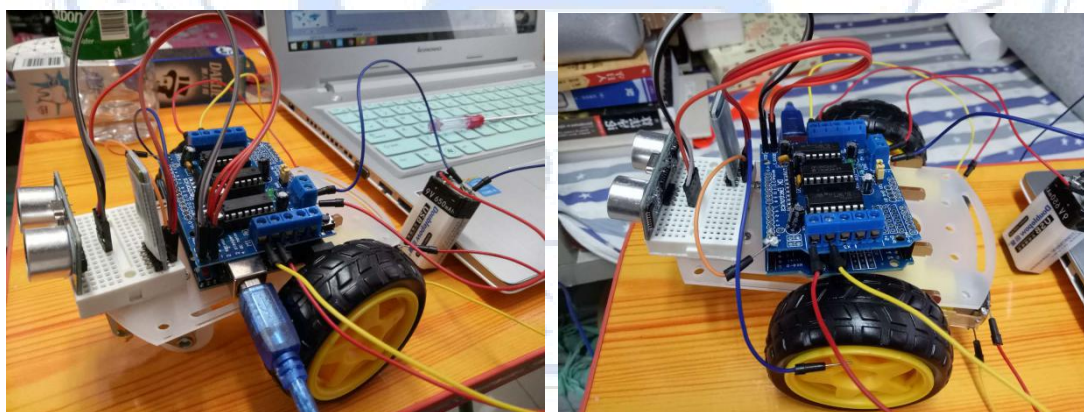
(4) 连接电路

蓝牙插件插在面包板上，共有四个引脚，两个引脚用于供电，两外两个用于传输信息。输入输出与 UNO 板输出输入口对应，剩余两个引脚与扩展板左上角相连。

超声波探测器插入在面包板上，四个引脚，GND 口接到扩展板 SER1 串口的负极，ECHO 口接到扩展板 SER1 串口的 S 极，TRIG 口接到扩展板 SER1 串口的 S 极，UCC 口接到扩展板 SER1 串口的正极。

我们将两个马达的正负极与 UNO 电机驱动扩展板 M1、M4 相连。

将蓄电池正负极分别与 UNO 电机驱动扩展板的 GND，+M 两个口相连接，不要连错。正常供电时，扩展板绿灯亮。



三、总结反思

- 1、我们开始想做四轮驱动小车，后面经过讨论，还是选择两轮驱动。两轮小车灵活轻便，在进行避障转向时效果会好一点。
- 2、我们考虑到橡皮筋固定小车不稳定，所以考虑使用支架固定小车。

3、我们配置环境开始编译时，编译报错：找不到头文件 **AFMotor.h**。下载后放在同一目录下即可。

4、组装完毕后发现小车没有动，我们开始调试小车。首先查找是不是马达或者电池的问题，我们单独用蓄电池为马达供电，发现马达正常工作，故排除这种可能。再与其他组交流后，发现我们电机驱动板绿灯不亮，所以发现是电机驱动板的问题。我们购置新的电机驱动板，但是目前尚未到货。

5、我们进一步确定是电机驱动板与蓄电池相连的串口坏了，所以用 **M1、M4** 串口连接蓄电池，发现小车可以跑。但是此时蓝牙发烫，并且蓝牙的灯没有亮，蓝牙被烧坏了。

6、**小车能够直走，并能正确躲避障碍物**。但是有时候由于接触不良以及摩擦，导致两个轮胎转速不同。

四、代码

```
#include <AFMotor.h>

AF_DCMotor Rback_motor(4);
AF_DCMotor Lback_motor(1);

float RLRatio = 1.0;

int maxSpeed = 80;//最大速度

char getstr;//串口数据

int IntervalTime;//测距时间

int trig = 9;//sero1 保留端口

int echo = 10;//sero2 保留端口

int control=1;//0:人工控制;1:自动控制
```

```
void setup() {  
  
    // put your setup code here, to run once:  
  
    Serial.begin(9600);  
  
    Rback_motor.setSpeed(0);  
  
    Rback_motor.run(RELEASE);  
  
    Lback_motor.setSpeed(0);  
  
    Lback_motor.run(RELEASE);  
  
    delay(2000);  
  
    //超声波测距  
    pinMode(echo, INPUT);  
    pinMode(trig, OUTPUT);  
}  
  
void loop() {  
  
    getstr = Serial.read();  
  
    driver();  
  
    if(control==1){  
  
        ultrasonicCar();  
  
    }  
  
}  
  
void forward() {
```



```
Rback_motor.run(BACKWARD);  
  
Lback_motor.run(BACKWARD);  
  
Rback_motor.setSpeed(maxSpeed*RLRatio);  
  
Lback_motor.setSpeed(maxSpeed);  
  
}
```

```
void backward() {  
  
    Rback_motor.run(FORWARD);  
  
    Lback_motor.run(FORWARD);  
  
    Rback_motor.setSpeed(maxSpeed*RLRatio);  
  
    Lback_motor.setSpeed(maxSpeed);  
  
}
```

```
void stopcar() {  
  
    Rback_motor.setSpeed(0);  
  
    Lback_motor.setSpeed(0);  
  
    Rback_motor.run(RELEASE);  
  
    Lback_motor.run(RELEASE);  
  
}
```

```
void left(){  
  
    Rback_motor.run(FORWARD);  
  
    Lback_motor.run(BACKWARD);
```

```
Rback_motor.setSpeed(maxSpeed*RLRatio);  
  
Lback_motor.setSpeed(maxSpeed);  
  
}
```

```
void right(){  
  
    Rback_motor.run(BACKWARD);  
  
    Lback_motor.run(FORWARD);  
  
    Rback_motor.setSpeed(maxSpeed*RLRatio);  
  
    Lback_motor.setSpeed(maxSpeed);  
  
}
```

```
void driver() {  
    if ((getstr == '<') && (RLRatio > 0.1)) {  
        RLRatio -= 0.1;  
        Serial.println(RLRatio);  
    }  
  
    if (getstr == '>') {  
        RLRatio += 0.1;  
        Serial.println(RLRatio);  
    }  
  
    if (getstr == '5') {  
        Serial.println("stopcar");  
    }  
}
```

```
stopcar();

control=0;

}

if (getstr == '1') {

    Serial.println("forward");

    forward();

    control=0;

}

if (getstr == '2') {

    Serial.println("backward");

    backward();

    control=0;

}

if (getstr == '3') {

    Serial.println("right");

    right();

    control=0;

}

if (getstr == '4') {

    Serial.println("left");

    left();

    control=0;

}
```

```
if (getstr=='6'){  
    control=1;  
    Serial.println("Auto Control On");  
}  
  
if(getstr=='7'){  
    control=0;  
    Serial.println("Auto Control Off");  
}  
  
if(getstr=='8'){  
    float s;  
    s = getdistance();  
    Serial.println(s);  
}  
}  
  
float getdistance() {  
    digitalWrite(trig, 1);  
    delayMicroseconds(15);  
    digitalWrite(trig, 0);  
    IntervalTime = pulseIn(echo, HIGH);  
    return IntervalTime / 58.00;  
}
```

```
void ultrasonicCar() {  
  
    float s = 0;  
  
    s = getdistance();  
  
    ultrasonicCar(s);  
  
}
```

```
void ultrasonicCar(float s) {  
  
    if ((s < 40)&&(s>0)){//40cm 以內  
  
        randomright();  
  
    }else{  
  
        forward();  
  
    }  
}  
  
void randomright(){  
  
    Rback_motor.run(BACKWARD);  
  
    Lback_motor.run(FORWARD);  
  
    Rback_motor.setSpeed(maxSpeed*RLRatio);  
  
    Lback_motor.setSpeed(maxSpeed);  
  
    long n = random(50,1000);//隨機轉向 0.05 - 1 秒  
  
    delay(n);  
  
    Rback_motor.setSpeed(0);  
  
    Lback_motor.setSpeed(0);  
  
}
```