**基于Arduino高速运转的循迹小车的制作**

**（黑线轨迹、速度达1.2m/s以上、PID控制、含源代码接线方法和调试方法）**

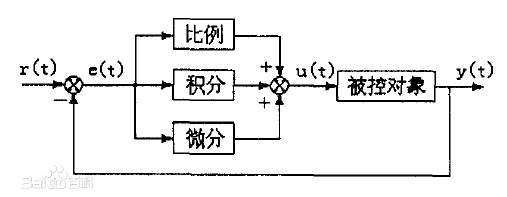
**作者：北辰远\_code**

**前言：**

循迹小车要达到高速运转利用高低电频或者pwm很难做到，只能达到循迹效果，并且当路况复杂时涉及到的情况很多，代码情况也很多，稍有考虑不周，会导致运行过程出错，同时还要花费大量时间进行调试。因此，制作循迹最重要的是自主的自动控制，而寻迹小车非常适合利用PID来控制。

何为PID控制？

PID控制又被称为比例（proportion）积分(integral)、微分(derivative)控制, 由于其算法简单、鲁棒性好、可靠性高，故被广泛用于工业过程控制中。下面，我们以循迹小车的运转来说明PID控制发挥的作用，循迹小车要沿着黑线前进，传感器检测到黑线就可知车的位置与黑线的误差。当仅考虑比例控制时，控制信号与误差成正比，当传感器在检测范围内时，小车距离黑线较远时误差较大，故控制量也会较大，小车则会调整前进方向，快速靠近黑线以减少误差，当小车中央足够靠近黑线时，误差也就足够小，控制信号也会很小，同时小车调整也会减小，使得小车维持在一定误差范围内运行。比例控制虽然可以快速减小误差，但最大的问题是当误差小于一定值时小车只能维持在一定的误差轨迹上运行，造成了所谓的稳态误差，这个误差不会随时间而被比例控制消除。所以此时引入积分控制，它会不断累积误差的值，使得我们的控制量增加，小车继续调整前进方向，最终在这一机制作用下成功消除稳态误差，达到既定轨迹，当然上述的模型有点太过于理想化，试想小车在直线轨迹高速运行，正前方有一个大转弯，此时车也正在保持原来的状态运行，则车相对既定轨迹急剧偏离，误差也随之快速增大，此时微分控制就可以大显神通了，微分控制器计算了误差的变化率，当小车急剧相对偏移时，微分控制量将随着误差的变化率的增大而增大，使得小车快速调整前进方向以回到原来的位置，进而使得小车稳定运行在既定轨道。（积分控制提供了惯性，微分控制提供了阻尼）



**PID控制原理图**

材料准备：

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **名称** | **数量** | **作用** |
| 1 | ArduinoUNO开发板 | 1 | 软硬件交互 |
| 2 | L298N电机驱动 | 1 | 驱动模块 |
| 3 | TCRT5000L五路红外循迹传感器 | 1 | 循迹 |
| 4 | 底板（建议自制） | 1 | 构架 |
| 5 | 12V/16V锂电池（含充电器） | 1 | 动力来源（提高电能） |
| 6 | 开关 | 1 | 控制电路通断 |
| 7 | 导线（公对公、母对母、公对母） | 若干 | ------------- |
| 8 | LM2569稳压 | 1（可有可无，为确保电池过放，建议使用） | 稳压模块 |

线路连接：

|  |  |
| --- | --- |
| **Arduino UNO** | **TCRT5000L** |
| GND | GND |
| 5V/3V(由于串口不够用建议接3V) | 5V |
| A5/19 | OUT1 |
| A4/18 | OUT2 |
| A3/17 | OUT3 |
| A2/16 | OUT4 |
| A1/15 | OUT5 |

|  |  |
| --- | --- |
| **Arduino UNO** | **L298N** |
| 5 | ENA |
| 10 | ENB |
| 6 | IN1 |
| 7 | IN2 |
| 8 | IN3 |
| 9 | IN4 |

|  |  |
| --- | --- |
| **电源** | **L298N** |
| + | 12V |
| - | GND |

|  |  |
| --- | --- |
| **Arduino UNO** | **L298N** |
| 5V | 5V |
| GND | GND |

|  |  |  |
| --- | --- | --- |
| **L298N** | 马达输出A | 马达输出B |
| **电机** | A | B |

**代码：**

float Kp , Ki = 0, Kd ; //pid弯道参数参数

float error = 0, P = 0, I = 0, D = 0, PID\_value = 0; //pid直道参数

float previous\_error = 0, previous\_I = 0; //误差值

static int initial\_motor\_speed = 113; //此处值为0-255的值，受电池电压影响，需要自己调试

int A[5] = {19, 18, 17, 16, 15};

double L, R;

float a, b, c, d, p, m;

int ena = 5;

int enb = 10;

void setup()

{

Serial.begin(9600);

pinMode(6, OUTPUT);

pinMode(7, OUTPUT);

pinMode(8, OUTPUT);

pinMode(9, OUTPUT);

pinMode(ena, OUTPUT);

pinMode(enb, OUTPUT);

pinMode(A1, INPUT);

pinMode(A2, INPUT);

pinMode(A3, INPUT);

pinMode(A4, INPUT);

pinMode(A5, INPUT);

}

void loop()

{

Kp = 5.75+0.125;

Kd = 23; //此处的Kp和Kd也需使用者根据车的结构和既定轨迹调试进行调试，

m = 0;

a = 3, b = 6, c = 11, d = 18, p = 1;

trac\_run();

}

void trac()

{

unsigned char temp = 0b00000; //临时变量用于新一轮采集

for (int i = 0; i < 5; i++)

temp |= digitalRead(A[i]) << i; //轮询5个传感器输出，并将查询结果转换为编码形式

switch (temp)

{

case 0b01111: error = -d; break;

case 0b00111: error = -c; break;

case 0b10111: error = -b; break;

case 0b10011: error = -a, m = 10; break;

case 0b11011: error = 0, m = 20; break;

case 0b11001: error = a, m = 10; break;

case 0b11101: error = b; break;

case 0b11100: error = c; break;

case 0b11110: error = d; break;

default: p = 0; break;

}

}

void pid()

{

P = error;

I = I + error;

D = error - previous\_error;

PID\_value = (Kp \* P) + (Ki \* I) + (Kd \* D);

previous\_error = error;

}

void trac\_run()

{

trac();

pid();

L = p \* ( initial\_motor\_speed + PID\_value + m);

R = p \* ( initial\_motor\_speed - PID\_value + m);

if (abs(L) > 255) L = L / abs(L) \* 255;

if (abs(R) > 255) R = R / abs(R) \* 255;

if (abs(L) == 0) L = 1;

if (abs(R) == 0) R = 1;

motor();

}

void motor()

{

if (L > 0 && R > 0)

analogWrite(enb, L),

analogWrite(ena, R),

motordir(0, 1, 1, 0);

else if (- L > 0 && - R > 0)

analogWrite(enb, -L),

analogWrite(ena, -R),

motordir(1, 0, 0, 1);

else if (- L > 0 && R > 0)

analogWrite(enb, -L),

analogWrite(ena, R),

motordir(0, 1, 0, 1);

else if (L > 0 && -R > 0)

analogWrite(enb, L),

analogWrite(ena, -R),

motordir(1, 0, 1, 0);

else

motordir(0, 0, 0, 0);

Serial.print("左轮");

Serial.print(L);

Serial.print("左轮");

Serial.println(R);

}

void motordir(int out1, int out2, int out3, int out4) //输入1 0，控制方向

{

digitalWrite(6, out1);

digitalWrite(7, out2);

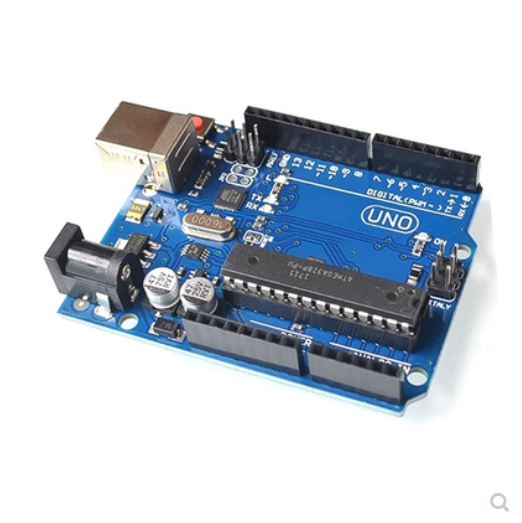
digitalWrite(8, out3);

digitalWrite(9, out4);

}

零件型号或名称：

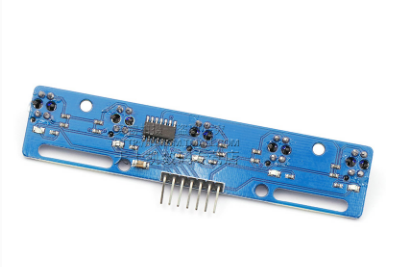
Arduino UNO 开发板：



驱动模块：

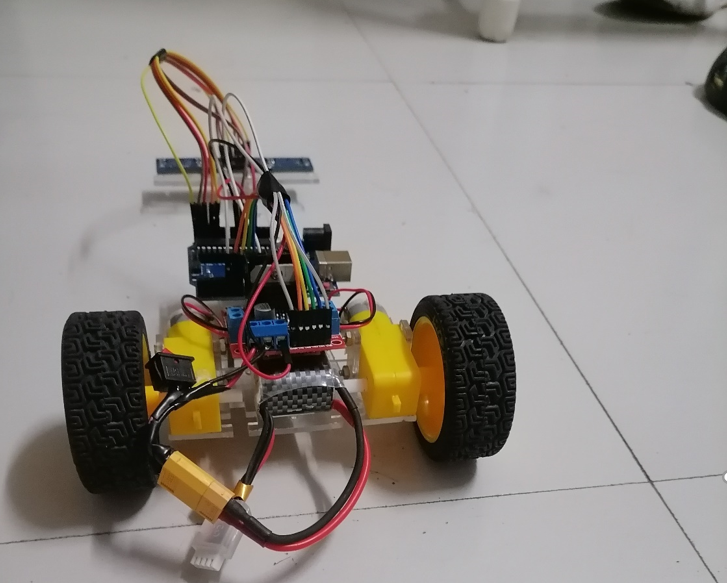
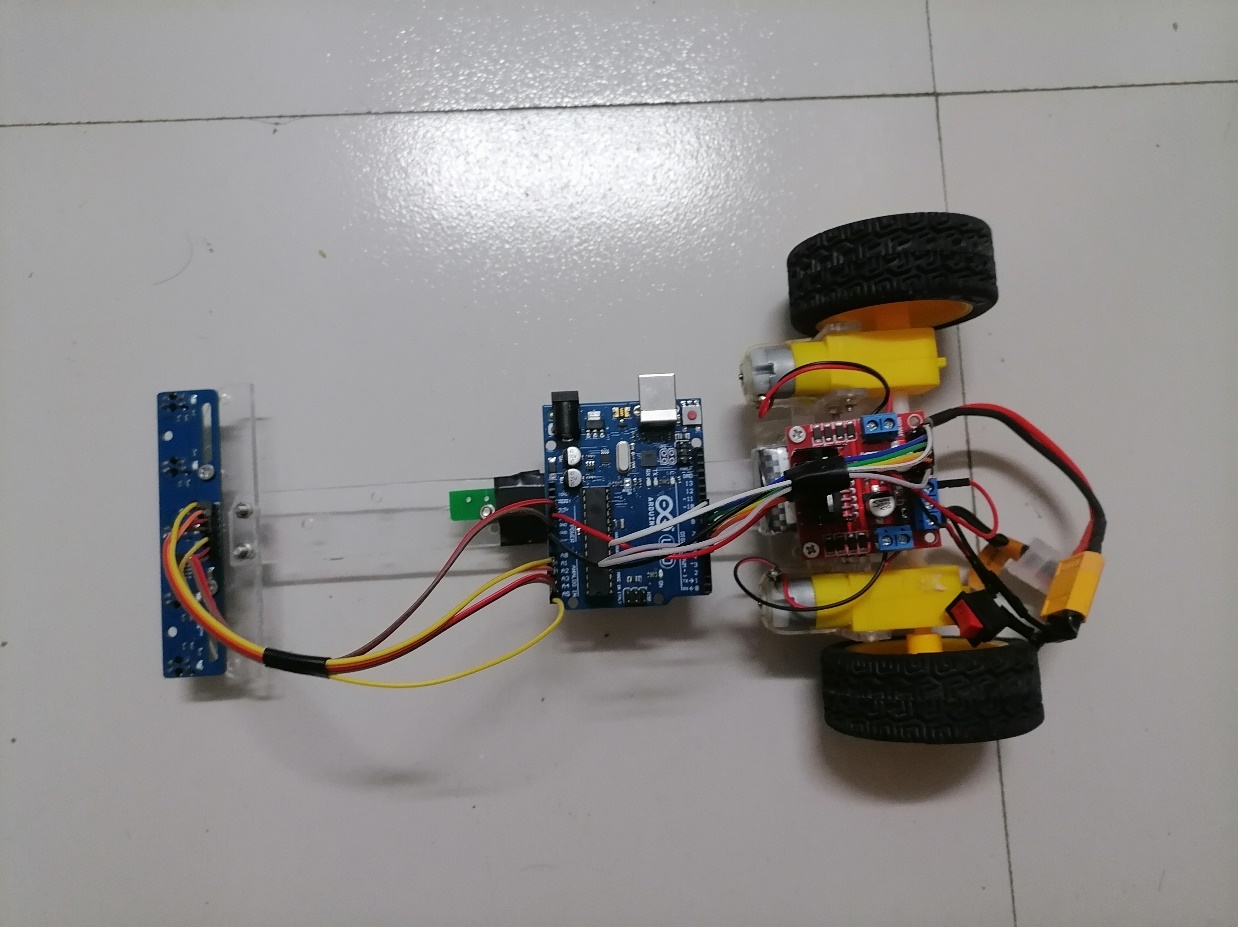
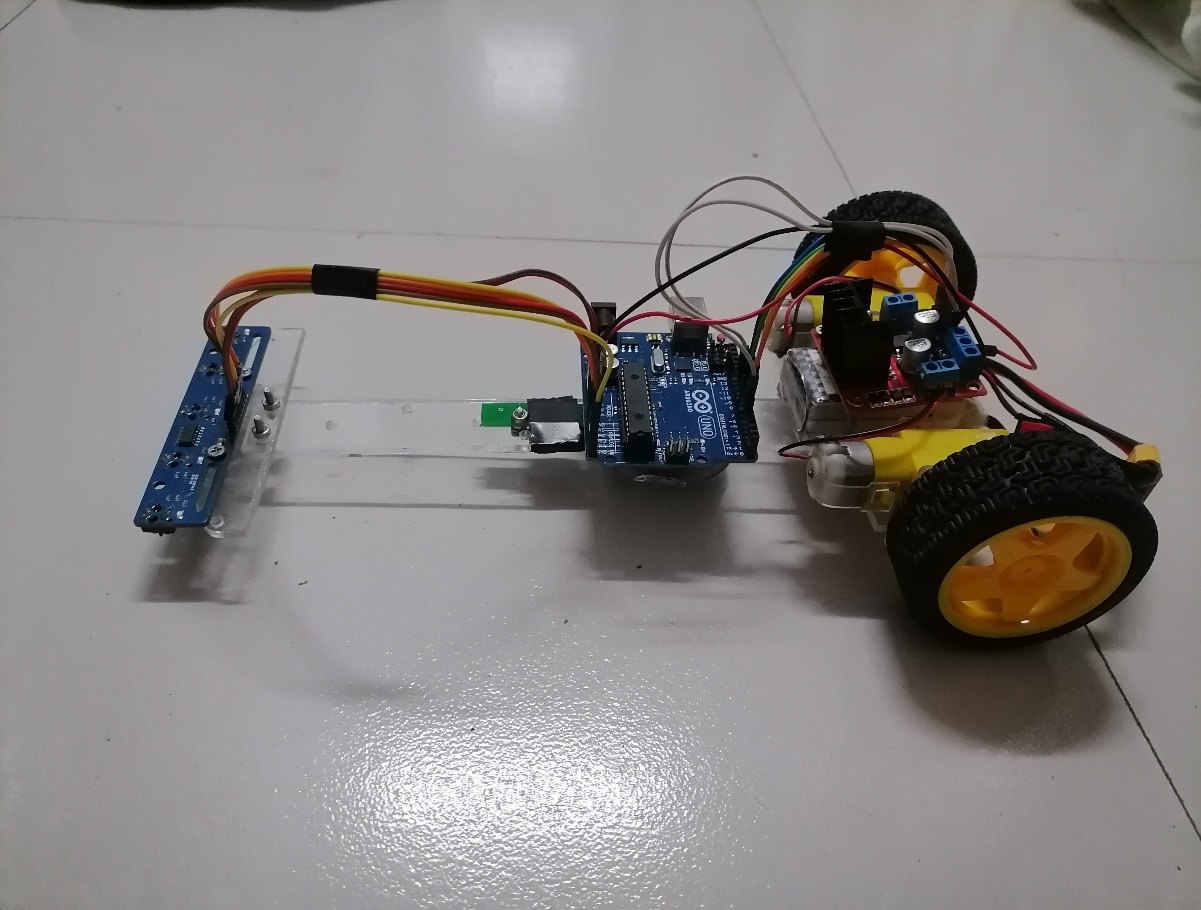


TCRT5000L五路红外循迹传感器：



电池和底板（本人自己切割的）不做展示，自行制作或购买。

成品展示：（本车已在尺寸为两个一米直线，加上两个直径一米的半圆，总长5.14m的赛道亲测）



个人建议：

如果想制作出高速运转的循迹小车尽量自己切割底板，底板结构也会影响小车速度，小车底板要低，前部支撑轮（不一定是轮，光滑支撑也可以）阻力不能太大，电池质量大，要放在两个后轮轴线正上方，电池较轻时要将重物置于两个后轮轴正上方，以增加小车的滑动摩擦力，同时两个后车轮要有粗糙度，以增大摩擦力，传感器距离前部支撑轮要稍远。