# 智能车驱动板部分说明文档

## 提示

- △ 在开发前请仔细阅读下面注意事项。
  - 1. 在开发之前请阅读资料文件夹【参考资料】→【智能车接线说明中文版.pdf】。
  - 2. 请将智能车放在平整的地面上运行, 防止底盘被刮坏。
  - 3. 智能车电池为可充电锂电池,其满电电压约为8V,为安全起见,请务必使用智能车配套的原装充电器为电池充电。
  - 4. 当电池电量不足时需关机充电。如长时间使用不充电会导致电池过放从而无法充电。当智能车后方的两个指示灯变为红色并闪烁说明电池电压较低,建议停止使用并充电; <u>小</u> 当蜂鸣器报警则说明电池电量严重不足,此时请务必关机并给电池充电!
  - 5. 在二次开发及调试的过程中,如果需要改变智能车运动状态(如在前进的过程中突然要求其后退),此时请务必先将车轮的速度清零(即让智能车停止运动),然后再对其运动状态进行改变。 比如上位机发送指令将智能车速度设为最快速度前进,此时如果想要为智能车发送全速后退指令, 需要先发送一个停止指令让智能车停下来,然后再发送后退指令让智能车全速后退。 // 注意,这 里很重要,桥电路需要有死区时间,否则可能会导致智能车损坏。

## 硬件连接

智能车接线图请参照【参考资料】→【智能车接线说明中文版.pdf】进行接线。

## LoRa 模块配置

如果需要用到手柄通讯或软串口,可使用LoRa模块进行通讯。LoRa模块在发货时默认已配置好。如需修改参数,请仔细阅读【LoRa配置软件】文件夹中的文档。

智能车底盘下图红框位置与LoRa Pin对Pin兼容,可直接将配置好的LoRa安装后使用。



# 开发板简介

在了解代码之前,首先我们要了解一下Arduino Nano控制板。

Arduino Nano控制板主要技术参数如下表所示:

MCU	Atmega328P
工作电压	5V
Flash Memory	32KB(其中有0.5KB用于系统引导)
SRAM	2KB (Atmega328P)
EEPROM	1KB (Atmega328P)
模拟输入引脚	8个
输入/输出引脚直流电流	40mA
输入电压	7-12V
数字输入引脚	22个
PWM引脚	6个
3.3V引脚电流	50mA
单板尺寸 (长×宽)	45mm×18mm
重量	≈7g
时钟频率	16MHz

下面是Arduino Nano的引脚说明图:

Arduino-Nano-Pinout

## 程序烧录方法

#### 第一步:下载Arduino IDE开发环境:百度或微软应用商店里搜索Arduino IDE就可以找到。

下载地址:

https://www.arduino.cc/en/Main/Software?setlang=cn

下载好后打开软件界面如下图所示:

》微信截图\_20210519104756

#### 第二步:下载并安装USB转串口驱动。

下载地址:

LINUX 驱动程序, 支持 32/64 位系统:

http://www.wch.cn/downloads/CH341SER\_LINUX\_ZIP.html

MAC OS 驱动程序, 支持 32/64 位系统:

http://www.wch.cn/downloads/CH341SER\_MAC\_ZIP.html

WINDOWS 驱动程序,支持 32/64 位 Windows 10/8.1/8/7/VISTA/XP , SERVER 2016/2012/2008/2003 , 2000/ME/98 :

http://www.wch.cn/downloads/CH341SER ZIP.html

安装方法:下载好后运行,单机"安装"。

過 驱动安装(X64) 驱动安装 <b>/</b> 卸载	×
选择INF文件:	CH341SER.INF ~
安装	WCH.CN
卸载	[]
帮助	

弹出"驱动安装成功"窗口即可,如下图所示:



#### 第三步:编写程序并下载

在编写程序之前,我们可以先验证一下开发板及编程环境,这个过程中也可以了解程序下载方法。

- 1. 将Arduino Nano开发板从智能车上取下并与电脑连接,如下图所示:
- 2. 打开Arduino IDE开发环境,依次点击:文件→示例→ 01.Basics → Blink

💿 sketch_ma	y19a   Arduino	1.8.13 (Windows Store 1.8.42.0)		-	$\times$
文件 编辑 项目	工具 帮助				
新建 打开… 打开最近的 项目文件夹	Ctrl+N Ctrl+O				₽. ▼
不例       关闭       保存       另存为       页面设置       打印       首选项       关闭	Ctrl+W Ctrl+S Ctrl+Shift+S Ctrl+Shift+P Ctrl+逗号 Ctrl+逗号 Ctrl+Q	△ 内置示例 01.Basics 02.Digital 03.Analog 04.Communication 05.Control 06.Sensors 07.Display 08.Strings 09.USB 10.StarterKit_BasicKit 11.ArduinoISP	<ul> <li>AnalogReadSerial BareMinimum</li> <li>Blink</li> <li>DigitalReadSerial Fade</li> <li>ReadAnalogVoltage</li> <li>&gt;</li> </ul>		^
		所有升发被的示例。 Bridge Ethernet Firmata LiquidCrystal SD Stepper Temboo TFT WiFi Arduino Nano 的例子 EEPROM SPI Wire			~

此时会弹出一个新的窗口,如下图所示:



3. 选择开发板型号。

### 依次点击:工具→开发板→ Arduino Nano,如下图所示:

0	Blink   Ardu	ino 1.8.13 (Windows Store 1.8.42.0)			-	$\times$
文件	編辑项目	工具 帮助				
		自动格式化	Ctrl+T			<b>.</b> 0-
		项目存档				
B	Blink	修正编码并重新加载				
1	/*	管理库	Ctrl+Shift+I			
2	Blink	串口监视器	Ctrl+Shift+M			
3		串口绘图器	Ctrl+Shift+L			
4 5	Turns	WiFi101 / WiFiNINA Firmware Updater	r	cond, repeatedly.		
6	Most A	开发板: "Arduino Nano"	;	开发板管理器		
7	it is	处理器: "ATmega328P (Old Bootloade	r)" >	Arduino Yún		
8	the co	端口: "COM3"	;	Arduino Uno		
9	If you	取得开发板信息		Arduino Duemilanove or Diecimila		
11	model,	编程器: "USBtinvISP"	;	Arduino Nano		
12	1100000.	條코리무程序		Arduino Mega or Mega 2560		
13	modifi	ed 8 May 2014		Arduino Mega ADK		
14	by Sco	tt Fitzgerald		Arduino Leonardo		
15	modifi	ed 2 Sep 2016		Arduino Leonardo ETH		
16	by Art	uro Guadalupi		Arduino Micro		
17	modifi	ed 8 Sep 2016		Arduino Esplora		
18	by Col	by Newman		Arduino Mini		
19				Arduino Ethernet		
20	This e	xample code is in the public do	main.	Arduino Fio		
21	http:/	(www.arduine.cc/on/mutorial/plic	n k	Arduino BT		
22	*/	/www.ardurno.cc/en/Tutofial/Bll	<u>115.</u>	LilyPad Arduino USB		
2.5	· ·			LilyPad Arduino		 _
				Arduino Pro or Pro Mini		
				Arduino NG or older		

4. 选择处理器型号。

依次点击:工具→处理器→ ATmega328P(Old Bootloader)

Blink | Arduino 1.8.13 (Windows Store 1.8.42.0)

文件	编辑 项目	I	- 帮助					
			自动格式化	Ctrl+T			2	o <u>.</u> .
			项目存档					
Blir	nk		修正编码并重新加载				5	2
1/	/*		管理库	Ctrl+Shift+I				^
2	Blink		串口监视器	Ctrl+Shift+M				
3	10 10 10 1 1 1 1		串口绘图器	Ctrl+Shift+L				
4	Turns		WiFi101 / WiFiNINA Firmware Updater		co	nd, repeatedly.		
6	Most	A	开发板: "Arduino Nano"	>	0	n the UNO, MEGA and ZERO		
7	it is		处理器: "ATmega328P (Old Bootloader)"	>		ATmega328P		
8	the c	0	端口: "COM3"	>	•	ATmega328P (Old Bootloader)		
9	If yo	u	取得开发板信息			ATmega168		
10	model	'						
11	<u>https</u>	:	编程音: "USBTINYISP"	,				
12			烧录引导程序					
13	modif	ied	8 May 2014					
14	by Sc	ott	Fitzgerald					
15	modif	ied	2 Sep 2016					
16	by Ar	turo	Guadalupi					
17	modif	ied	8 Sep 2016					

 $\Box$   $\times$ 

5. 选择端口号。鼠标右键点击"计算机",选择"管理",在弹出的菜单里面选择"设备管理器",右侧菜单 点击"端口"前面的三角符号,找到 USB-SERIAL CH340 ,后面括号 COM 口编号即是端口号,如: 下图后面括号内为COM3,则端口号为COM3。

接下来我们回到Arduino IDE开发环境,依次点击:工具→端口→ COM3(根据开发者计算机实际端口 号自行选择),如下图所示:

🥯 B	💿 Blink   Arduino 1.8.13 (Windows Store 1.8.42.0) – 🗆								
文件:	编辑 项目	工具	帮助						
			自动格式化	Ctrl+T					
			项目存档						
Blir	nk		修正编码并重新加载						
1 /	*		管理库	Ctrl+Shift+I					
2	Blink		串口监视器	Ctrl+Shift+M					
3			串口绘图器	Ctrl+Shift+L					
4 5	Turns		WiFi101 / WiFiNINA Firmware Updater			cond, repe	atedly.		
6	Most A		开发板: "Arduino Nano"		>	On the UN	O, MEGA and Z	ERO	
7	it is		处理器: "ATmega328P (Old Bootloader)"		>	6. LED_BU	ILTIN is set	to	
8	the co		端口: "COM3"		>	串行端口			
9	If you		取得开发板信息			COM1	on your Ardu	ino	
11	model,		编程器·"IISBtinyISD"		,	COM2			
12	<u>nccps.</u>		烧寻引导程序			COM3			
13	modifi	ed 8	May 2014						
14	by Sco	tt E	Titzgerald						
15	modifi	ed 2	2 Sep 2016						
16	by Art	uro	Guadalupi						
1.0		1 0	0.0010						

#### 6. 选择编程器

依次点击: 工具→编程器→ USBtinyISP



#### 7. 程序编译

点击程序编译按钮"√",如果程序没有问题,软件会提示"编译完成",如下图所示:

#### 8. 程序烧录

点击"上传"按钮,等待上传成功,如下图所示:

由于我们烧录的是示例程序,此时可以验证一下开发板是否工作正常,若开发板上的红色指示灯"L"闪烁,则说明程序烧录成功。

提示:若烧录过程中界面一直停留在"上传",或显示"上传失败",可以尝试烧录过程中按住开发板上的 轻触开关,等到状态从"正在编译项目..."变为"上传..."时松开轻触开关即可。

#### 智能车驱动程序说明

#### 驱动电路简介

在了解智能车驱动程序之前,我们首先来了解一下L298N电机驱动模块。

L298N是一款经典的直流有刷电机驱动芯片,其本质上是使用H桥电路驱动。L298N可以驱动两个二相电机,也可以驱动一个四相电机,可以直接用IO口提供信号,而且电路简单,使用方便。



L298N的引脚图、内部电路及其他详细参数这里就不做赘述,用户可上网自行查询或在资料文件夹中" datasheet"中查阅,此处简单介绍一些L298N的逻辑功能表,如下表所示:

IN1	IN2	EN	电机状态
Х	Х	0	停止
1	0	1	顺时针
0	1	1	逆时针
0	0	0	停止
1	1	0	停止

由上表可知: EN为低电平时, 输入电平对电机控制起作用, 当ENA为高电平, 输入电平为一高一低时, 电机正转或反转。同为低电平电机停止, 同为高电平电机刹停。

L298N电路去如下图所示:



上图是从智能车原理图中截取出来的,由于L298N可以驱动两个二相电机,所以这里不仅又IN1、IN2,还有IN3和IN4。并且又两个使能口(ENA、ENB)。从图中可以看出,我们是通过D10、D9、D7、D6提供信号,D11、D5为使能管脚。所以我们在设计程序时需要控制这几个IO口的信号。另一路L298N电路与之相同,只是提供信号的管脚有所不同。

# 代码简介

在打开代码之前,我们首先回顾一下刚刚打开Arduino IDE时的界面:

```
sketch_may19a | Arduino 1.8.13 (Windows Store 1.8.42.0)
                                                                             Х
文件 编辑 项目 工具 帮助
sketch may19a
 1 void setup() {
 2
    // put your setup code here, to run once:
 3
 4 }
 5
 6 void loop() {
 7
    // put your main code here, to run repeatedly:
 8
 9}
```

我们可以看这里有一个"void setup()"和一个"void loop()"顾名思义, setup是用来写配置参数的, 例如 我们定义的一些变量、对IO口的配置都可以放在这里, loop则相当于while(1)。 打开智能车驱动板部分资料文件夹→代码→CarCtrl\_4

浏览代码, Arduino IDE使用的是C语言, 有C语言基础的应该很容易理解代码含义。

```
1| #include <SoftwareSerial.h>//包含SoftwareSerial库2| SoftwareSerial mySerial(3,4);//设置软串口的RX、TX
```



从电路图中我们可以看出,LoRa模块的发送和接收连接在Arduino Nano的D3和D4管脚,这里就要将D3、D4管脚作为一组软串口来使用。所以要包含"SoftwareSerial.h"库文件。

接下来是定义外设端口号,例如:

其他的定义与之类似,这里就不一一列举,用户可参照原理图来了解每个管脚的定义。

接下来是"void setup()":

```
      38| Serial.begin(115200);
      //串口波特率设置为115200(如没有用到Arduino Nano的

      硬件串口可以不做设置)
      //软串口波特率设置为115200

      39| mySerial.begin(115200);
      //软串口波特率设置为115200

      40|
      //软串口波特率设置为115200

      41| pinMode(Buzzer,OUTPUT);
      //将Buzzer管脚(D8)设置为输出模式

      42| pinMode(LeftFront_1,OUTPUT);
      //将LeftFront_1管脚(D7)设置为输出模式

      43| pinMode(LeftFront_2,OUTPUT);
      //将LeftFront_2(D6)设置为输出模式

      44| pinMode(enA,OUTPUT);
      //将enA管脚(D5)设置为输出模式
```

在分析代码前,我们先来回顾一下PWM的概念。

PWM (脉冲宽度调制技术) 是一种对模拟信号电平进行数字编码的方法,我们可以采用高分辨率计数器,根据调制方波占空比的方法来对一个具体的模拟信号的电平进行编码。PWM信号仍然是数字信号。 通的时候即是直流供电被加到负载上的时候,断的时候即是供电被断开的时候。只要带宽足够,任何模拟值都可以使用PWM 进行编码。

先定义一个255级PWM,用于智能车调速:

34| int TotalWidth = 255;

PWM部分代码如下:

73 void	<pre>PWMCtrl(int tFirst,int tSecond, int tThrid, int tFourth){</pre>
74	
75	digitalWrite(enA,HIGH);  //将ENA设置为高
76	digitalWrite(enB,HIGH);  //将ENB设置为高
77	digitalWrite(enC,HIGH);  //将ENC设置为高
78	digitalWrite(enD,HIGH);  //将END设置为高
79	for(unsigned char i=0;i <totalwidth;i++)< td=""></totalwidth;i++)<>
80	{
81	if(tFirst>255)
82	tFirst = 255;
83	if(tSecond>255)
84	tSecond = 255;
85	if(tThrid>255)
86	tThrid = 255;
87	if(tThrid>255)
88	tFourth = 255;
89	<pre>delayMicroseconds(1);</pre>
90	if(i==tFirst) //根据写入的参数将ENA设置为低,下同。
91	digitalWrite(enA,LOW);
92	if(i==tSecond)
93	digitalWrite(enB,LOW);
94	if(i==tThrid)
95	<pre>digitalWrite(enC,LOW);</pre>
96	if(i==tFourth)
97	<pre>digitalWrite(enD,LOW);</pre>
98	}
99 }	

然后我们就可以根据L298N的逻辑表对智能车的运动状态进行设计,例如前进的代码如下所示:

100   ヽ 进	<pre>void GoForward(int tFirstF, int tSecondF,</pre>	<pre>int tThridF, int tFourthF){</pre>	//前
101			
102			
103	digitalWrite(LeftFront_1,LOW);	//将D7设置为低	
104	digitalWrite(LeftFront_2,HIGH);	//将D6设置为高	
105			
106	<pre>digitalwrite(LeftBehind_1,LOw);</pre>	//将D10设置为高	
107	digitalWrite(LeftBehind_2,HIGH);	//将D9设置为高	
108			

```
109
         digitalwrite(RightFront_1,LOW);
                                                  //将A3设置为高
110
         digitalWrite(RightFront_2,HIGH);
                                                   //将A4设置为高
111
         digitalWrite(RightBehind_1,LOW);
112
                                                  //将A1设置为高
113
         digitalwrite(RightBehind_2,HIGH);
                                                   //将A2设置为高
114
         PWMCtrl(tFirstF,tSecondF,tThridF,tFourthF); //设置PWM
115
116
         }
```

设置好后,我们可以尝试在void loop()中写下如下代码,烧录后观察智能车的运动状态:

```
void loop(){
    GoForward(220,220,220,220);
}
```

当我们将GoForward () 函数中的参数值变小, 会发现车的速度也随之变慢。

```
void loop(){
    GoForward(200,200,200,200);
}
```

在CarCtrl\_4程序中,已经将基本的前进、后退、左转、右转和停止的函数写好,用户可根据需求自行修改速度和方向参数。

接下来我们来看串口通讯代码,以硬件串口为例, 软串口与之相同:

```
void SerialCtrl()
{
   while (Serial.available() > 0)
                                        //串口有数据可度吗?
   {
       comdata += char(Serial.read()); //将comdata写入
       delay(2);
   }
   if (comdata.length() > 0)
                                          //如果comdata长度大于0
    {
       if (comdata == "DOA"){
                                          //如果comdata为"DOA",则Flag_Set = 1 ,
下同
           Flag_Set = 1;
           }
         else if (comdata == "DOB"){
           Flag_Set = 2;
           }
         else if (comdata == "DOC"){
           Flag_Set = 3;
           }
         else if (comdata == "DOD"){
           Flag_Set = 4;
           }
         else if (comdata == "DOE"){
           Flag_Set = 5;
           }
         else if (comdata == "DOH"){
           Flag_Set = 6;
           }
```

```
else if (comdata == "DOI"){
    Flag_Set = 7;
    }
    NoResponse();
    comdata = ""; //清空comdata
}
```

上面的代码中的Flag\_Set可以在"FlagCtrl()"函数中找到。其代码如下:

```
void FlagCtrl(){
     if(Flag_Set == 1){
                                 //当Flag_Set值为1
   GoForward(235,235,235,235); //执行前进(速度为235,最大255),下同
   }
   else if(Flag_Set == 2){
     GoBack(235,235,235,235);
     }
   else if(Flag_Set == 3){
     TurnLeft(235,235,235,235);
     }
   else if(Flag_Set == 4){
     TurnRight(235,235,235,235);
     }
   else if(Flag_Set == 5){
     StopRun();
     }
   else if(Flag_Set == 6){
     LeftCorrect(225,225,225,225);
     }
   else if(Flag_Set == 7){
     RightCorrect(225,225,225,225);
     }
 }
```

可以看出这是一个控制智能车运动状态的函数。结合"SerialCtrl()"函数可知,当串口收到字符串"DOA"智能车会执行前进动作,下同。

这里需要注意的是:该代码是为了配合QA200DK的路标识别Demo程序使用的,故忽略状态位问题,用 户可根据自己的通信协议自行修改串口通讯函数及智能车运动状态。

串口相关指令:

- 前进: DOA
- 后退: DOB
- 左转: DOC
- 右转: DOD

停止: DOE

⚠ 上文中已经提及,在控制过程中,切勿让智能车某一状态时突然改变运动状态。应在中间加上一个停止指令,让智能车先停止运动,否则可能会导致智能车损坏。例如为智能车发送"DOA"指令让智能车前进,此时想让智能车后退或转弯,请不要直接发送"DOB"指令,先发送一个"DOE",再发送"DOB"即可。

```
void loop(){
   SerialCtrl();
   SoftSerialCtrl();
   FlagCtrl();
   LowPower();
   BuzzerBeep();
}
```

//串口通讯函数 //软串口通讯函数 //转台控制函数 //电压监测函数 //蜂鸣器低压报警函数