

01

第一章 MAX的行走路线



DFROBOT
DRIVE THE FUTURE

前情回顾

在前面的上手指南中，我们已经对Arduino的下载和安装，程序的编译和下载有了简单的了解。

本章内容

编写程序让MAX机器人的行驶路线为正方形。

本章知识点

1. 程序主要分几个部分；
2. 函数的定义；
3. setup()函数和loop()函数；
4. 注释的使用；
5. 延时函数的使用；



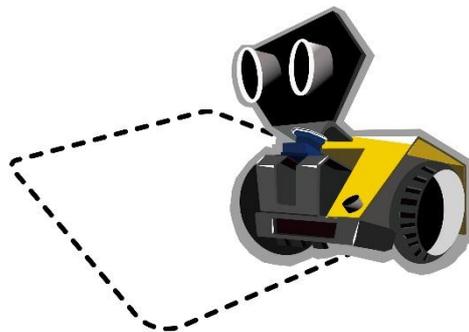
一、编写及下载程序

打开 **ArduinoIDE**，将下面的代码输入到编辑区中

```
/*  
程序功能：让 MAX 机器人的行驶路线为正方形。  
作 者：DFRobot  
*/  
#include <DFRobot_MAX.h>           //载入 MAX 驱动库  
DFRobot_MAX myMax;                 //载入 MAX 驱动程序  
  
void setup() {  

```

点击“验证”按钮编译程序。用 **USB 线**将 MAX 和电脑连接在一起，点击“上传”按钮将程序上传到 MAX 机器人。打开电源开关，我们就能看见，MAX 向前直行 2 秒后，又向右转。就这样一直不断的循环循环啊。



二、程序的主要组成部分及函数的定义

MAX 程序的主要组成部分

回顾我们的程序结构，主要分**驱动库**和**驱动函数**；`void setup()`函数 `void loop()`函数。

1

载入 MAX 驱动库和驱动程序，想要驱动 MAX，在程序开头必须有这条语句，否则机器人将无法工作，程序编译也会出错的。

```
#include <DFRobot_MAX.h>
DFRobot_MAX myMax;
```

2

在 `setup` 函数中初始化 MAX 系统。

```
void setup() {
  myMax.begin();
}
```

3

在 `loop` 函数中，设置 MAX 的行驶路线或功能

```
void loop() {
  myMax.forward(100,100);
  .....
}
```

函数的定义

函数通常为具有一个个功能的小模块，通过这些功能的整合，就组成了我们的整段代码，一个完整的功能实现。这些功能块也能被反复运用。这时，就体现函数的好处了。在程序运行过程中，有些功能可能会被重复使用，所以只需程序中调用一下函数名就可以了，无需重复编写。而 `setup()` 和 `loop()` 比较特殊，一段代码中只能使用一次。除此之外还有一个概念需要我们了解，那就是函数的返回值，比如“`void`”就是**函数无返回值**的号，并且后面的**括号内为空**，这个我们以后会经常用到。你是否对函数有了一个简单的概念了呢？不明白也没关系，在我们之后的项目还会涉及到的。

注意：

这几项在 MAX 的代码中可是**缺一不可**的

```
#include <DFRobot_MAX.h>
DFRobot_MAX myMax
void setup()
myMax.begin()
void loop()
```

三、代码回顾

注释的作用

代码的开头如下所示：

```
/*  
程序功能：让 MAX 机器人的行驶路线为正  
方形。  
作 者：DFRobot  
*/
```

另外还有一种注释方式，如下所示：

```
//载入 MAX 驱动库
```

这就是代码中的说明文字，也可以叫做注释，`/*.....*/`可以注释多行，在`/*`和`*/`中间的所有内容都将被编译器忽略，不进行编译。

这是另外一种注释方式，是以“`//`”开头的，这个符号所在行之后的文字都将不被编译器编译。

注释在代码中很重要的，他可以帮助我们理解代码。在比较复杂的项目中，代码就发挥了很大的作用了。这样可以很方便的理解代码了。

在 ArduinoIDE 中会自动的把注释的文字颜色变为灰色。

代码分析

在程序中我们最先载入 MAX 的驱动库和驱动程序：

```
#include <DFRobot_MAX.h> //载入 MAX 驱动库  
DFRobot_MAX myMax; //载入 MAX 驱动程序
```

没有这两项，会编译不过。这样 MAX 就接受不了指令了。库文件点击下面的下载链接就可以获取了。

[\[MAX 库文件下载链接\]](#)

接下来是 `setup()` 函数：

```
void setup() {  
  myMax.begin();  
}
```

↓ 只执行一次

`setup` 在英文中是“设置”的意思。顾名思义，`setup()` 函数是用于一些初始化的设置：

```
myMax.begin(); //初始化 MAX。
```

`setup()` 函数主要目的是在 `loop()` 函数运行之前为程序做一些必要的设置。在 Arduino 中首先调用的就是 `setup()` 函数，每次在 Arduino 上电或者重启后，`setup()` 函数只运行一次。从“`{`”开始到“`}`”结束。

再接着下面就进入了我们的主函数 `loop()` 函数

```
void loop() {  
  myMax.forward (100,100);  
  .....  
}
```



循环执行

看字面意思知道该函数是在程序运行过程中不断的循环，`loop()` 函数中的每条语句都是依次进行的，一直到函数的最后，然后在进入 `loop` 函数的第一条语句再次开始，这样一直的循环下去，知道关闭 Arduino 或者重启。

在 `setup()` 函数中初始化 MAX 后，就开始执行 `loop()` 函数。

Arduino 程序必须包含 `setup()` 和 `loop()` 两个函数，否则不能正常工作。

在本章，我们希望 MAX 的行驶路线为正方形，那么在程序中是怎样实现的呢？

我们先来看 `loop()` 函数里的第一条语句：`myMax.forward (100,100)`；这行语句的意思是告诉 MAX：你要按左轮 100，右轮 100 的速度前进。没接到其他指令前，就这样一直行驶。

随后，我们使用了延时函数：`delay(2000)`；结合上一句，这两行语句加一起的意思是告诉 MAX：你按 100 的速度前进，2 秒钟后，我可能会给你新的任务。

2 秒钟到了，新的任务来了：`myMax.forward (100, 0)`；告诉 MAX：你按左轮 100，右轮 0 的速度右转前进。延时 1.5 秒钟后，直行。就这样一直循环执行 `loop()` 函数里面的语句就能实现 MAX 的行驶路线成正方形了。

课后思考

1、去掉延时，MAX 会怎么样？

前面我们学习了延时，它的主要功能是等待一段时间后执行下一条语句，这样就能看见 MAX 直行 2 秒后就向右转。那如果不使用延时函数：`delay()`，会怎么样？同样是刚才的程序，去掉它的延时（`delay()`）看看。

2、尝试更改 `loop` 主循环内的代码和参数，让机器人 MAX 行走更加多变的路线。

1)、尝试更改延时函数 `delay(ms)` 的参数值大小，来改变前进、右转的时间。

2)、尝试更改前进函数 `myMax.forward (L,R)` 的参数（L 和 R 的值），来改变行驶的速度和转弯的角度。

3)、尝试在 `loop` 主循环中加入后退函数 `myMax.backward (L,R)` 让机器人增加后退的动作。