14

第十四章

声控机器人

DFROBOT
DRIVE THE FUTURE

前情回顾

在上个章节里，我们学习了如何使用GoBLE软件控制MAX的行走方式。

本章内容

学习声音传感器的原理及函数的使用，并用声音控制MAX前进、后退、右转、停止。

本章知识点

1. 学习声音传感器的原理；
2. 学习什么是子函数；
3. 学习如何自定义子函数；
4. 学习如何使用声音传感器的函数；

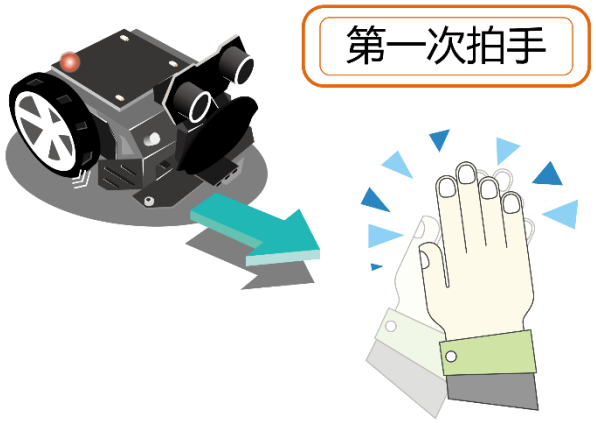
一、 编写并下载程序

打开 ArduinoIDE，将下面的代码输入到编辑区中

```
/*
  程序功能：声控机器人，实现的是利用拍掌来控制 MAX 的行驶状态；拍一次掌前进，再拍一次掌
            后退，然后再拍一次就右转等功能。
  作 者：DFRobot
*/
#include <DFRobot_MAX.h>
DFRobot_MAX myMax;
int command,i=1;                //定义两个变量，用来判断 MAX 应该前进还是后退

void setup() {
   myMax.begin();
   Serial.begin(9600);
}

void MicVal()                   //声音传感器检测的函数
{
   int micVal;                  //定义一个变量用来存放声音传感器的数据
   do{
      delayMicroseconds(10);                          //消除误差
      micVal=myMax.micValue();          //读取声音传感器的数据并赋值给 micVal
   }while( micVal<700 || micVal == -1 ); //如果声音小于 700 的话或者为-1 的话，就为误判断
   command=i;                          //将 i 得到的值赋值给 command
   return ;                            //终止一个函数，并返回函数的结果值
}

void loop() {
       MicVal();       //调用声音传感器的函数，有击掌声音的时候，才退出函数
    switch(command){
       case 1:   myMax.forward(120,120);          //  如果 command 为 1，MAX 以 120 的速度直行前进
                 myMax.rgbLed(0, 255, 0, 0);   //0 号灯亮红色
                 i++;                                 // i+1
                 break;
```

```
        case 2:   myMax.backward(120,120);        // 如果 command 为 2，MAX 以 120 的速度后退
                  myMax.rgbLed(1, 0, 255, 0);   //1 号灯亮绿色
                  i++;
                  break;
        case 3:   myMax.forward(200,100);     //   如果 command 为 3，MAX 右转，400ms 后直行
                  myMax.rgbLed(2, 0, 0, 255);delay(400);     //2 号灯亮蓝色
                  myMax.forward(120,120);
                  i++;
                  break;
        case 4:   myMax.forward(0,0);           // 如果 command 为 4，MAX 停止
                  for(int a=0;a<=4;a++){
                  myMax.rgbLed(a,0, 0, 0);
                  } i=1;                          //给 i 这个变量赋值 1
                  break;
    }
}
```

上传成功后拨开 MAX 的开关；拍一次手掌，MAX 就改变一次状态。第一次拍手 MAX 直行，0 号 RGB 灯亮红色；第二次拍手 MAX 后退，1 号 RGB 灯亮绿色；第三次拍手，MAX 右转，3 号 RGB 灯亮蓝色；第四次拍手，MAX 停止，并将 i 置 1；如果接下来再拍一次手掌，MAX 则直行，程序就这样一直循环。



二、 声音传感器的原理

MIC 声音传感器是一款基于麦克风的声音传感器，能检测周围环境中声音的强度，然后通过反馈的电压值来体现声音的大小。从右图的串口监视器中知道声音越大，反馈的数值越大。

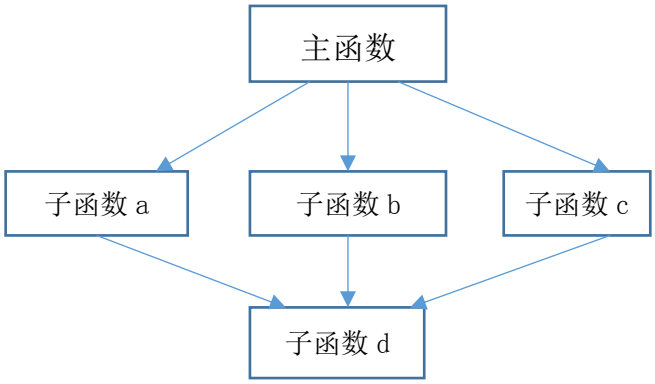
三、 子函数

在第一章我们学习过函数的定义，了解了函数通常为一个个功能的小模块，通过这些模块的整合来实现一个较大的程序。

在一个完整的代码中，一般分主函数和子函数两种；

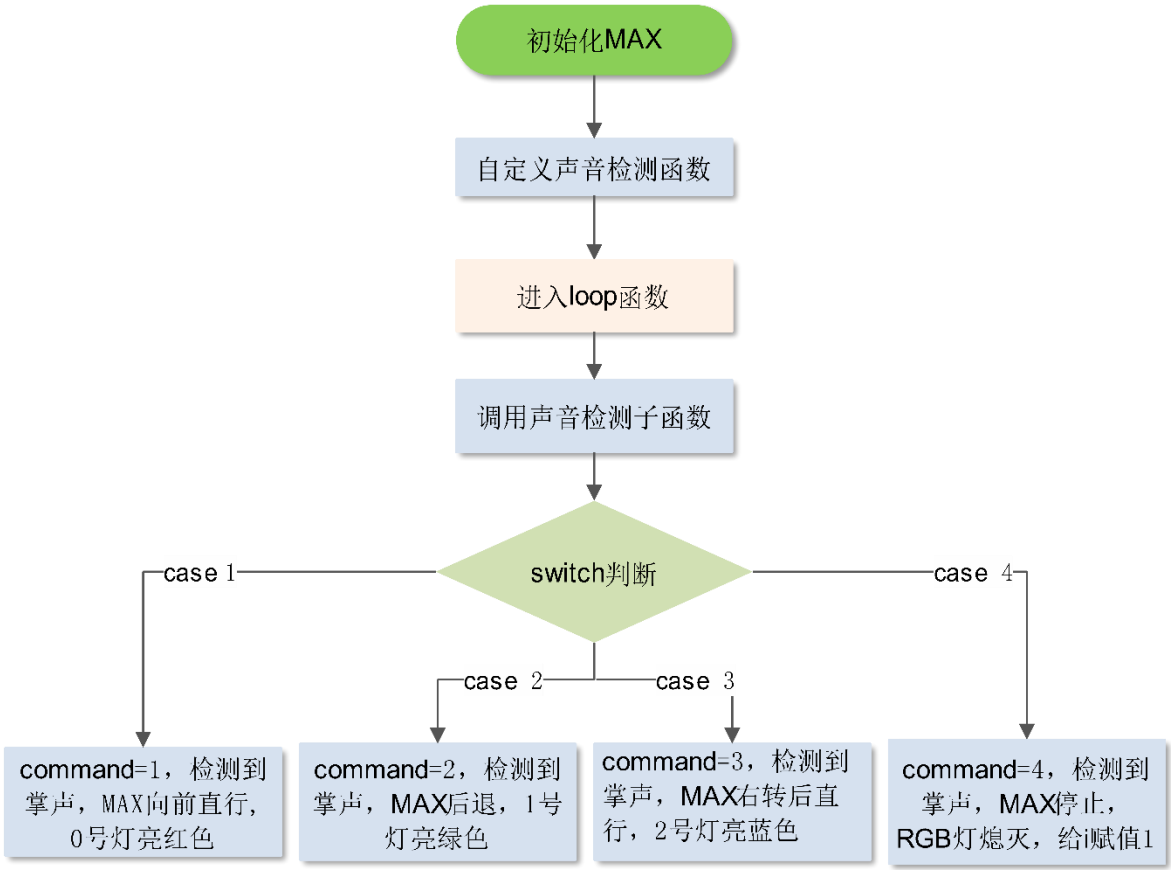
主函数：为程序的入口函数，程序执行的时候最先调用的函数就是这个函数了。在 arduino IDE 中通常以 loop()函数为主函数。

子函数：用户可以自定义的函数，可以被主函数调用，也可以被其他子函数调用的。其实子函数里的代码也可以直接写在主函数中，但是为了代码的可读性。一般会使用子函数将不同功能区分开。



四、 代码分析

本章主要学习如何自定义子函数和如何使用声音控制 MAX 行驶，其程序流程图如下图。



代码分析：

代码中，首先定义两个变量：

```
int command,i=1;
```

用来判断 MAX 行驶的状态。定义 i 的初值为 1

自定义声音传感器检测函数 MicVal()；首先先声明函数：

```
void MicVal()
```

void 是函数无返回值的信号，其后面的括号内为空。

子函数内的程序结构和正常的程序结构一样，它的作用只是将每一个功能模块化，这样增加了程序的可读性。代码例程：

```
void MicVal()
 {   int micVal;
    do{
      delayMicroseconds(10);
      micVal=myMax.micValue();
   }while( micVal<700 || micVal == -1 );
   command=i;
   return ;
}
```

子函数的功能是检测声音传感器的数据；自定义好函数后，直接在 loop 函数里调用子函数 MicVal() 这样就避免了在 loop 函数里书写代码。

调用子函数的格式：

```
void loop() {
   MicVal();
      ……
 }
```

直接在此处键入子函数的名称，即 void 后面的函数名，以 “;” 结尾

子函数内代码分析：

在函数 MicVal()内，首先定义一个变量：

```
int micVal;
```

用来存放声音传感器的数据值。

利用 while 循环判断 ,当声音小于 700 或者为-1，就为误判，则一直执行 do…while…里面的语句：

```
delayMicroseconds(10);     //延时 10us
 micVal=myMax.micValue();
```

其中 delayMicroseconds 为一种延时函数 ,单位 us。

声音传感器函数：myMax.micValue();是用来读取声音传感器的值。

如果声音大于 700 则跳出 while 循环，执行：

```
command=i;
 return;
```

其中 return 表示终止一个函数，并向被调函数返回一个值。如果函数执行不需要返回计算结果，也经常需要返回一个状态来表示函数执行的是否顺利，通常的状态码为 -1，0，1。主函数可以通过返回值判断被调用函数的执行情况。

return 函数使用格式：

return ; 表示返回一个空值。 return 0; 表示正常终止函数。

return 1; 表示非正常终止函数 return i; 表示返回一个值为 i。

return -1; 表示返回一个代数，一般用在子函数结尾，表示非正常终止函数。

进入 loop 函数后，首先调用声音传感器函数：

```
MicVal();
```

调用声音函数，每次有击掌声的时候，就退出函数，然后执行下一段函数。利用 switch 控制函数完成。

代码例程：

```
switch(command){
  case 1:   myMax.forward(120,120);          // 如果 command 为 1，MAX 以 120 的速度直行前进
            myMax.rgbLed(0, 255, 0, 0);   //0 号灯亮红色
            i++;                                  // i+1
            break;
            .......
  case 4:   myMax.forward(0,0);            // 如果 command 为 4，MAX 停止
            for(int a=0;a<=4;a++){
            myMax.rgbLed(a,0, 0, 0);
            } i=1;                              //给 i 这个变量赋值 1
            break;
}
```

声音传感器每检测到一次声音，MAX 就变换一种行驶状态。同时 i 的值也自加 1。如第一次检测到声音时，i 置 1，MAX 直行；第二次检测到声音时，i 置 2，MAX 后退；第三次检测到声音时，i 置 3，MAX 右转；第四次检测到声音时，i 置 4，MAX 停止，然后给 i 赋值 1，当再次检测到声音时，i 置 1，MAX 前进，然后就这样一直循环下去。