

15

第十五章

按键的拓展—— 功能切换



DFROBOT
DRIVE THE FUTURE

前情回顾

在上个章节里，我们学习了声音传感器的原理及其函数的使用。

本章内容

学习时间函数，利用按键切换追光，蓝牙控制，巡线等功能。

本章知识点

1. 学习什么是时间函数；
2. 学习怎么用按键切换功能；



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：利用按键切换追光，蓝牙控制，巡线等功能。
作者：DFRobot
*/
#include <DFRobot_MAX.h> //载入 MAX 驱动库
#include <GoBLE.h> //载入 GoBLE 库
DFRobot_MAX myMax; //载入 MAX 驱动函数
unsigned long modeTime ;
int selectMode;

void rayTracing(){ //追光机器人
    while(1){
        int data_L = myMax.lightValue(1); //左光敏传感器数据
        int data_R = myMax.lightValue(0); //右光敏传感器数据
        if (data_R- data_L > 350) { //光线在左，向左移动
            myMax.forward(50,150);
        } else if (data_L- data_R > 350) { //光线在右，向右移动
            myMax.forward(150, 50);
        } else if (data_R + data_L < 1000) { //光线在前方，向前移动
            myMax.forward(100,100);
        } else { //未找到光线
            myMax.forward(0,0); //MAX 停止移动
        } if(myMax.buttonValue()==0){ //如果检测到按键被按下，退出循环，切换到下一个功能
            break;
        }
    }
}

void bluetoothControl(){ //蓝牙控制
    int up,down,left,right,rotate,stop; //定义 6 个按键
    while(1){
        if (Goble.available()){ //如果接收到的蓝牙数据为有效数据，就将接收到的数据赋值给各个按键变量
            up = Goble.readSwitchUp(); //前进
            down = Goble.readSwitchDown(); //后退
            left = Goble.readSwitchLeft(); //左转
            right = Goble.readSwitchRight(); //右转
            rotate = Goble.readSwitchSelect(); //旋转
            stop = Goble.readSwitchStart(); //停止
            if (up == PRESSED){ //如果上键被按下的话，MAX 前进
```

```

myMax.forward(120,120);
}else if(down == PRESSED){ //否则，判断下键是否被按下，按下，就后退
myMax.backward(120, 120);
}else if(left == PRESSED){ //否则，判断左键是否按下，按下，就左转
myMax.forward(100,200);
}else if(right == PRESSED){ //否则，判断右键是否按下，是，就右转
myMax.forward(200,100);
}else if(rotate == PRESSED){ //否则，判断旋转键是否按下，是，就旋转
myMax.forward(0,120);
}else if(stop == PRESSED) //否则，判断停止键是否按下，是，就停止
myMax.forward(0,0);
}if(myMax.buttonValue()==0){ //如果检测到按键被按下，退出循环，切换到下一个功能
break;
}
}
}

```

```

void linePatrol(){ //巡线机器人
int L,M,R;
int command; //定义 4 个变量
while(1){
R=myMax.lineValue(0); //读取右侧巡线传感器的值
M=myMax.lineValue(1); //读取中间巡线传感器的值
L=myMax.lineValue(2); //读取左侧巡线传感器的值
myMax.lineLED(1); //点亮巡线灯
if((L==1 && M==1 && R==1)|| (L==0 && M==1 && R==0)) command=1;
if((L==0 && M==1 && R==1)|| (L==0 && M==0 && R==1)) command=2;
if((L==1 && M==0 && R==0)|| (L==1 && M==1 && R==0)) command=3;
switch(command){
case 1:
myMax.forward(100,100); //当 command=1，MAX 向前直行
break;
case 2:
myMax.forward(150,70); //当 command=2，MAX 向前右转，并执行以下代码
if(L==0 && M==0 && R==0){ //如果三个巡线传感器都没有检测到黑线
myMax.playSound(10); //MAX 播放 10 号音效
myMax.forward(150,70); //MAX 继续右转，检测黑线
} break;
case 3:
myMax.forward(70,150); //当 command=3，MAX 向前左转，并执行以下代码
if(L==0 && M==0 && R==0){ //如果三个巡线传感器都没有检测到黑线
myMax.playSound(10); //MAX 播放 10 号音效
myMax.forward(70,170); //MAX 继续左转，检测黑线
} break;
}if (myMax.buttonValue()==0){ //如果检测到按键被按下，退出循环，切换到下一个功能

```

```

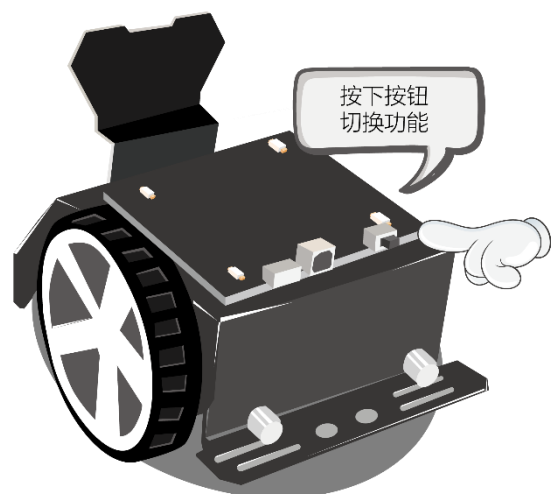
        myMax.lineLED(0);
        break;
    }
}
}

void setup(){
    myMax.begin();
    Serial.begin(115200);    //初始化串口并设置波特率为 115200
}

void loop(){
    if(millis()-150>modeTime){    //检测一次按键是否被按下，并且距离上一次按下后有 150ms 的等待时间
        if (myMax.buttonValue()==0){    //如果按键被按下
            selectMode++;    //选择模式加 1
            if(selectMode==4) selectMode=1;    //如果当选择模式等于 4 时，选择模式被置 1
            while (myMax.buttonValue()==0) delay(5);    //按下检测延时 5us
        }
        modeTime = millis();    //记录上一次按键被按下的时间
    }
    switch(selectMode){
        case 1: myMax.showFace(1,7);    //检测到按键被按下，显示白色的 1 号表情
                rayTracing();    //执行追光程序
                break;
        case 2: myMax.showFace(2,4);    //检测到按键被按下，显示蓝色的 2 号表情
                bluetoothControl();    //执行蓝牙控制程序
                break;
        case 3: myMax.showFace(3,3);    //检测到按键被按下，显示绿色的 3 号表情
                linePatrol();    //执行巡线程序
                break;
        default:
            break;
    }
}
}

```

上传完毕后，拨开 MAX 的开关，第一次按下按键表情板会显示白色的 1 号表情并执行追光程序；在按一次按键退出追光功能，执行蓝牙控制程序，显示蓝色的 2 号表情；按第三次按键退出蓝牙控制功能，执行巡线程序，显示绿色的 3 号表情。



二、代码回顾

首先载入所需要的库以及驱动函数，然后定义变量：

```
unsigned long modeTime ;
int selectMode;
```

modeTime 用来存放时间；selectMode 用来选择功能：

```
selectMode=1, 执行的是追光程序。
selectMode=2, 执行蓝牙控制程序。
selectMode=3, 执行的是巡线程序。
```

接着自定义三个子函数：

```
void rayTracing(){
    //追光功能程序
}
void bluetoothControl(){
    //蓝牙控制功能程序
}
void linePatrol(){
    //巡线功能程序
}
```

主要是实现追光、蓝牙控制、巡线功能。

注：在每一段的子函数里都用了 `while(1)` 死循环；当遇到按键被按下的时候，才跳出循环。

避免多余的资源浪费。

死循环：一个靠自身控制无法终止的程序。

```
例程：while(1){
    ...
    if(外部中断条件){
        break ; } //跳出循环
}
```

功能切换；利用时间函数 `millis()` 每隔 150ms 检测一次按键的状态来实现的。

millis()时间函数：

该函数可以获取机器运行的时间长度，单位为 ms，最长的记录时间接近 50 天，超过这个时间就要从 0 开始计时。

millis()函数的时间类型为 unsigned long，用 int 保存时间会得到错误的结果。

因为 int 型只能存储 32 秒 (32768 毫秒=32 秒)，超过这个时间就会出现变量溢出，所以会造成错误的结果。

代码例程：

```
if(millis()-150>modeTime){
    .....
}
modeTime = millis();
}
```

代码中的 modeTime 最开始是不存放任何数值的，后面将 `millis()` 函数得到的值赋值给它，这样它就会随着 `millis()` 值的变化而变化了。通过 `millis()` 函数不断记录时间，以此来判断两次按键的间隔时间是否 > 150ms，如果没有 > 150ms 就不做反应。因为按键被按下的时候会有抖动的过程，所以设置间隔时间来消除抖动。

在第三章按键的原理中知道 MAX 身上的按键属于低电平有效，所以是 `myMax.buttonValue()==0` 有效。

代码例程：

```
if (myMax.buttonValue()==0) { //如果按键被按下
    selectMode++;           //选择模式加 1
}
```

每当检测到按键被按下（**选择模式加 1**），就切换到下一个功能。

代码例程：`switch(selectMode){`

```
    case 1:
        // selectMode=1,执行追光
        break;
    case 2:
        // selectMode=2,执行蓝牙
        break;
    case 3:
        // selectMode=3,执行巡线
        break;
    default:
        break;
}
```

如果要用按键切换更多的功能，只需要将功能自定义成子函数，然后在 `loop` 函数中调用。

注：如果切换 4 种功能，需要将代码中的 `if(selectMode==4) selectMode=1;` 数字“4”更改为“5”，依次类推。