

为Arduino编写函数库

本文介绍如何为Arduino编写函数库。本文从一个简单的摩斯电码例子开始，描述如何将它的功能改写成函数库。这将使代码变得更为易用，也更易维护和升级功能。

先从一个简单的摩斯电码程序开始：

```
int pin = 13;

void setup()
{
    pinMode(pin, OUTPUT);
}

void loop()
{
    dot(); dot(); dot();
    dash(); dash(); dash();
    dot(); dot(); dot();
    delay(3000);
}

void dot()
{
    digitalWrite(pin, HIGH);
    delay(250);
    digitalWrite(pin, LOW);
    delay(250);
}

void dash()
{
    digitalWrite(pin, HIGH);
    delay(1000);
    digitalWrite(pin, LOW);
    delay(250);
}
```

运行以上程序，Arduino的13脚LED将按SOS方式（一种求救信号格式）闪烁。

这段代码中的一部分可以写成库函数：首先，用于闪烁的blink()和dash()两个功能函数；其次，用于指定使用哪个管脚的ledPin变量最后，初始化管脚的pinMode()函数调用

让我们把这段代码改写成函数库

一个函数库应至少包含两个文件：头文件（扩展名为.h）和源代码文件（扩展名为*.cpp*）**。

- 头文件包含函数库的声明，即函数库的功能说明列表；
- 源代码文件包含函数库的实际实现。

让我们来为这个函数库起个名字——“Morse”，那么头文件就命名为Morse.h。看看这个文件里都有些什么内容。

头文件的核心内容，是一个封装了成员函数与相关变量的类声明：

```
class Morse
{
public:
    Morse(int pin);
    void dot();
    void dash();
private:
    int _pin;
};
```

简单点说，类就是一个把函数和变量放在一起的集合。类里的函数与变量，其访问权限可以是 public（公有，即提供给函数库的使用者使用），也可以是 private（私有，即只能由类自己使用）。类有个特殊的函数—构造函数，它用于创建类的一个实例。构造函数的类型与类相同，且没有返回值。

头文件里还有些其它杂项。如为了使用标准类型和Arduino语言的常量，需要#include语句（Arduino的IDE会自动为普通代码加上这些#include语句，但不会自动为函数库加）。这些#include语句类似：

```
#include "Arduino.h"
```

最后，为了防止多次引用头文件造成各种问题，我们常用一种看起来有点奇怪的方式来封装整个头文件的内容：

```
#ifndef Morse_h
#define Morse_h
// the #include statment and code go here...
#endif
```

该封装的主要作用是防止头文件被引用多次。

通常也会在函数库的头文件里，加上一些关于作者、用途、日期、协议等注释。

最终完成的头文件如下：

```

/*
Morse.h - Library for flashing Morse code.
Created by Damo wang, November 20, 2015.
Released into the public domain.
*/

#ifndef Morse_h
#define Morse_h

#include "Arduino.h"

class Morse
{
public:
    Morse(int pin);
    void dot();
    void dash();
private:
    int _pin;
};

#endif

```

接下来，让我们继续完成源代码文件，Morse.cpp。

首先仍然是一些#include语句。这些语句让下面的程序能够使用Arduino的标准函数和刚才在Morse.h里声明的类。

```

#include "Arduino.h"
#include "Morse.h"

```

接下来是构造函数。再次说下，构造函数是当创建类的一个实例时调用的。在本例中，用于指定使用哪个管脚。我们把该管脚设置成输出模式并且用一个私有成员变量保存起来，以备其它函数使用。

```

Morse::Morse(int pin)
{
    pinMode(pin, OUTPUT);
    _pin = pin;
}

```

这段代码看起来有好几个怪地方。一是函数名之前的Morse::。这其实是用来指定该函数是Morse类的成员函数。下面定义类的其它成员函数时，将会一再出现。另一个不常见的是私有成员变量名_pin中的下划线。其实你可以按C++的命名规则，给它任意命名。加下划线是一种约定俗成的不成文规范，让我们既能区分传进来的pin参数，也能清晰地知道它的private私有性质。

下面终于到了我们转换实际代码的时刻了。除了Morse::和_pin，下面的代码与之前看起来没啥两样：

```

void Morse::dot()
{
    digitalWrite(_pin, HIGH);
    delay(250);
    digitalWrite(_pin, LOW);
    delay(250);
}

void Morse::dash()
{
    digitalWrite(_pin, HIGH);
    delay(1000);
    digitalWrite(_pin, LOW);
    delay(250);
}

```

好，让我们看看Morse.cpp全貌吧：

```

/*
    Morse.cpp - Library for flashing Morse code.
    Created by Damo wang, November 20, 2015.
    Released into the public domain.
*/

#include "Arduino.h"
#include "Morse.h"

Morse::Morse(int pin)
{
    pinMode(pin, OUTPUT);
    _pin = pin;
}

void Morse::dot()
{
    digitalWrite(_pin, HIGH);
    delay(250);
    digitalWrite(_pin, LOW);
    delay(250);
}

void Morse::dash()
{
    digitalWrite(_pin, HIGH);
    delay(1000);
    digitalWrite(_pin, LOW);
    delay(250);
}

```

这就是所有要做的工作（当然还有一些别的选项，容后再述）。来看看怎么使用这个函数库吧。

- 在你的代码库目录里（sketchbook中的libraries目录），创建一个叫Morse的子目录
- 把Morse.h和Morse.cpp拷贝到Morse目录里
- 启动Arduino IDE，打开Sketch > Import Library菜单，此时应能看到Morse菜单项。

本函数库将与使用它的代码一起编译。若编译不成功，请确认这些文件的扩展名是.cpp和.h，而不是.pde或.txt之类。

让我们用刚创建的函数库来重写SOS程序吧：

```
#include <Morse.h>

Morse morse(13);

void setup()
{
}

void loop()
{
  morse.dot(); morse.dot(); morse.dot();
  morse.dash(); morse.dash(); morse.dash();
  morse.dot(); morse.dot(); morse.dot();
  delay(3000);
}
```

与之前的程序相比，有一些不同：

加了一个#include语句。这条语句让程序可以使用Morse库且包含了对应的代码，最终下载到Arduino板上。若不再需要某个函数库，则应删除对应的#include语句，以减少生成的程序大小，节约空间。

创建了Morse类的一个实例：morse:

```
Morse morse(13);
```

当这行代码被执行时（实际上这行代码在setup函数之前执行），将传入参数（本例中的参数是13）并调用Morse类的构造函数。

注意setup()函数现在里面没语句了，因为原先对pinMode()的调用已经转移到函数库的构造函数里了。

最后，为了调用dot()和dash()成员函数，需要在之前加上morse.这样的前缀——需要使用的实例名。可以在程序里定义多个实例，每个实例拥有各自的管脚（保存在每个实例自己的_pin变量里）。调用某个实例的成员函数，使用的就是该实例的成员变量。如下例：

```
Morse morse(13);
Morse morse2(12);
```

在调用morse2.dot()时，_pin为12。

新程序里关于Morse函数库的内容无法被IDE识别和高亮。很不幸的是，目前Arduino的IDE无法自动识别自定义函数库里的内容（这个功能应该有），所以需要给Arduino一点小帮助：在Morse的文件夹里，创建一个叫keywords.txt文件，内容如下：

```
Morse      KEYWORD1
dash       KEYWORD2
dot        KEYWORD2
```

每一行均由关键字的名字、TAB键（非空格）、关键字种类顺序组成。类名是KEYWORD1，将被高亮成橘黄色；函数名是KEYWORD2，将被高亮成棕色。重启Arduino IDE后这些设置将生效。

若在函数库的文件夹内包含一些使用例子，会让库更易使用。只要在Morse文件夹内创建examples目录，然后把刚才写的程序（命名为SOS程序）文件保存到该目录即可。你可以通过Sketch > ShowSketch Folder菜单快速找到对应库函数的目录。重启Arduino IDE后，将会在**File > Sketchbook > Examples**菜单里找到Library-Morse例子。最好再加一些注释来说明如何使用你编写的函数库。

完成上述工作之后，可以把你的代码，完整的上传到你的github，分享给他人使用。