# 关于IRDuino

IRDuino是xxx.

# IRDuino使用

这里我们以一个电视机遥控器来模拟键盘的上下左右为例，说明IRDuino是如何使用的。下面是我的遥控器，可以看到中间刚好有上下左右的四个按键。我们让他和键盘的上下左右箭头对应起来。

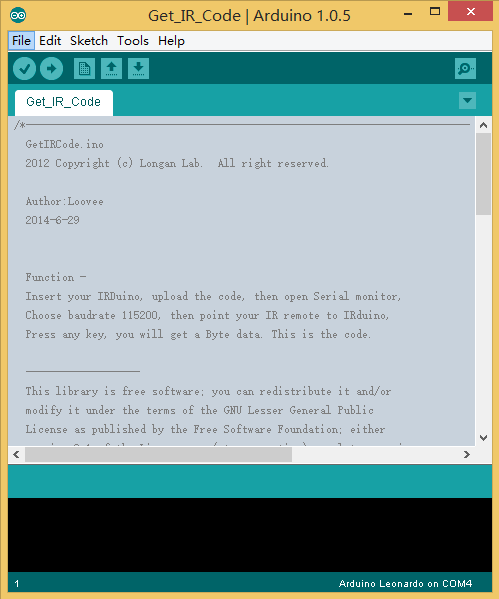


## 下载代码

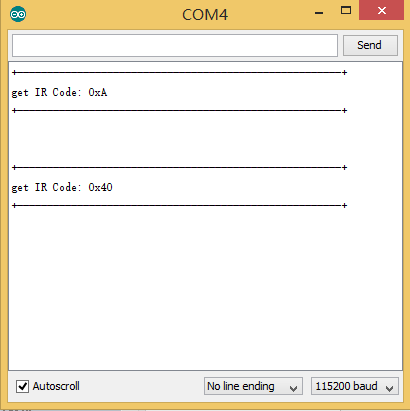
可以通过[这里](https://github.com/loovee/IRDuino)下载源代码，这是一个Arduino Library，下载后放到你的Arduino Libraries文件夹。如果解压后发现文件夹含有”-master”的字符串，请把该字符串删掉，这是github加上去的字符串，Arduino库的文件夹如果含有’-’会造成编译错误。

## 读取遥控器键值

IRDuino可以支持市面上大部分的红外遥控器。打开Get\_IR\_Code例程，可以读取红外遥控器某个按键的值。



Tool -> boards选择Arduino Leonardo，选择正确的COM，把代码下载到IRDuino。打开串口调试助手，并且对准IRDuino按下一个按键：



可以看到串口打印出了某个按键的键值，是一个八位的数据，大小从0~0xFF。从上图可以看到，我按下了两个按键，分别是0x0A和0x40。接着，我们用这种方式读出上下左右四个按键所对应的值：

|  |  |
| --- | --- |
| UP | 0xEE |
| DOWN | 0xAE |
| LEFT | 0xCE |
| RIGHT | 0x8E |

记下这几个值，下一步需要用到。

## 编写代码

打开Task\_Mode\_UDLR的example.

UDLR的意思是Up, Down, Left和Right。首先我们大致的讲解一下里面的代码。

// IR CODE DEFINE

#define IR\_CODE\_UP 0xEE

#define IR\_CODE\_DOWN 0xAE

#define IR\_CODE\_LEFT 0xCE

#define IR\_CODE\_RIGHT 0x8E

可以看到，我们定义了几个宏，用来标记上一步读到的IR值。

接着是setup里面的：

// add task item

iRduino.addItem(IR\_CODE\_UP, task\_up);

iRduino.addItem(IR\_CODE\_DOWN, task\_down);

iRduino.addItem(IR\_CODE\_LEFT, task\_left);

iRduino.addItem(IR\_CODE\_RIGHT, task\_right);

这里是添加了几个任务，添加的时候需要带入两个参数，一个是IR\_Code，一个是task函数指针。如果你对函数指针不熟悉，没有关系，实际上你在第二个参数填入函数名就可以了。这几个函数会在下面定义:

void task\_up()

{

iRduino.keyPressRelease(KEY\_UP\_ARROW);

}

void task\_down()

{

iRduino.keyPressRelease(KEY\_DOWN\_ARROW);

}

void task\_left()

{

iRduino.keyPressRelease(KEY\_LEFT\_ARROW);

}

void task\_right()

{

iRduino.keyPressRelease(KEY\_RIGHT\_ARROW);

}

从函数名不难看出这四个函数的功能。分别是按下Up, Down, Left以及Right这几个键。暂时不需要理会函数里面的内容，稍后我们会讲到。

从上面可以看到，要添加一个动作，只需要三步：

1. 记下读到的IR Code, 为了便于阅读，推荐用一个宏来记录
2. 写一个任务函数，用来执行一定的动作
3. 通过addItem函数，使该IR Code与上面的动作关联起来

## 如何写一个动作

首先我们看看上面见过的一个动作：

void task\_right()

{

iRduino.keyPressRelease(KEY\_RIGHT\_ARROW);

}

里面只有一句：iRduino.keyPressRelease(KEY\_RIGHT\_ARROW). 这个函数的原型是：

// press a key, then release it

void IRDuino::keyPressRelease(unsigned char keyNum)

这个函数实现的功能是按下一个按键(键盘的)，延时10ms，然后松开。另一个疑惑是，KEY\_RIGHT\_ARROW是什么。这个是UP那个按键对应的键值。其他按键的定义附录A。

还有其他的函数可供使用：

* 按下一个按键

void press(unsigned char keyNum);

* 松开所有按键

void releaseAll();

* 输入一段文字

void enterString(char \*str);

如果你使用过Windows操作系统的话，你应该不会对Ctrl+Alt+Del陌生。如果我们要写这样一个动作，可以这样实现：

void task\_logout()

{

iRduino.press(KEY\_LEFT\_CTRL); // press ctrl

iRduino.press(KEY\_LEFT\_ALT); // press alt

delay(10);

iRduino.press(KEY\_DELETE); // press Del

delay(10);

iRduino.releaseAll(); // release All

}

1. 按键键值

#define KEY\_LEFT\_CTRL 0x80

#define KEY\_LEFT\_SHIFT 0x81

#define KEY\_LEFT\_ALT 0x82

#define KEY\_LEFT\_GUI 0x83

#define KEY\_RIGHT\_CTRL 0x84

#define KEY\_RIGHT\_SHIFT 0x85

#define KEY\_RIGHT\_ALT 0x86

#define KEY\_RIGHT\_GUI 0x87

#define KEY\_UP\_ARROW 0xDA

#define KEY\_DOWN\_ARROW 0xD9

#define KEY\_LEFT\_ARROW 0xD8

#define KEY\_RIGHT\_ARROW 0xD7

#define KEY\_BACKSPACE 0xB2

#define KEY\_TAB 0xB3

#define KEY\_RETURN 0xB0

#define KEY\_ESC 0xB1

#define KEY\_INSERT 0xD1

#define KEY\_DELETE 0xD4

#define KEY\_PAGE\_UP 0xD3

#define KEY\_PAGE\_DOWN 0xD6

#define KEY\_HOME 0xD2

#define KEY\_END 0xD5

#define KEY\_CAPS\_LOCK 0xC1

#define KEY\_F1 0xC2

#define KEY\_F2 0xC3

#define KEY\_F3 0xC4

#define KEY\_F4 0xC5

#define KEY\_F5 0xC6

#define KEY\_F6 0xC7

#define KEY\_F7 0xC8

#define KEY\_F8 0xC9

#define KEY\_F9 0xCA

#define KEY\_F10 0xCB

#define KEY\_F11 0xCC

#define KEY\_F12 0xCD