

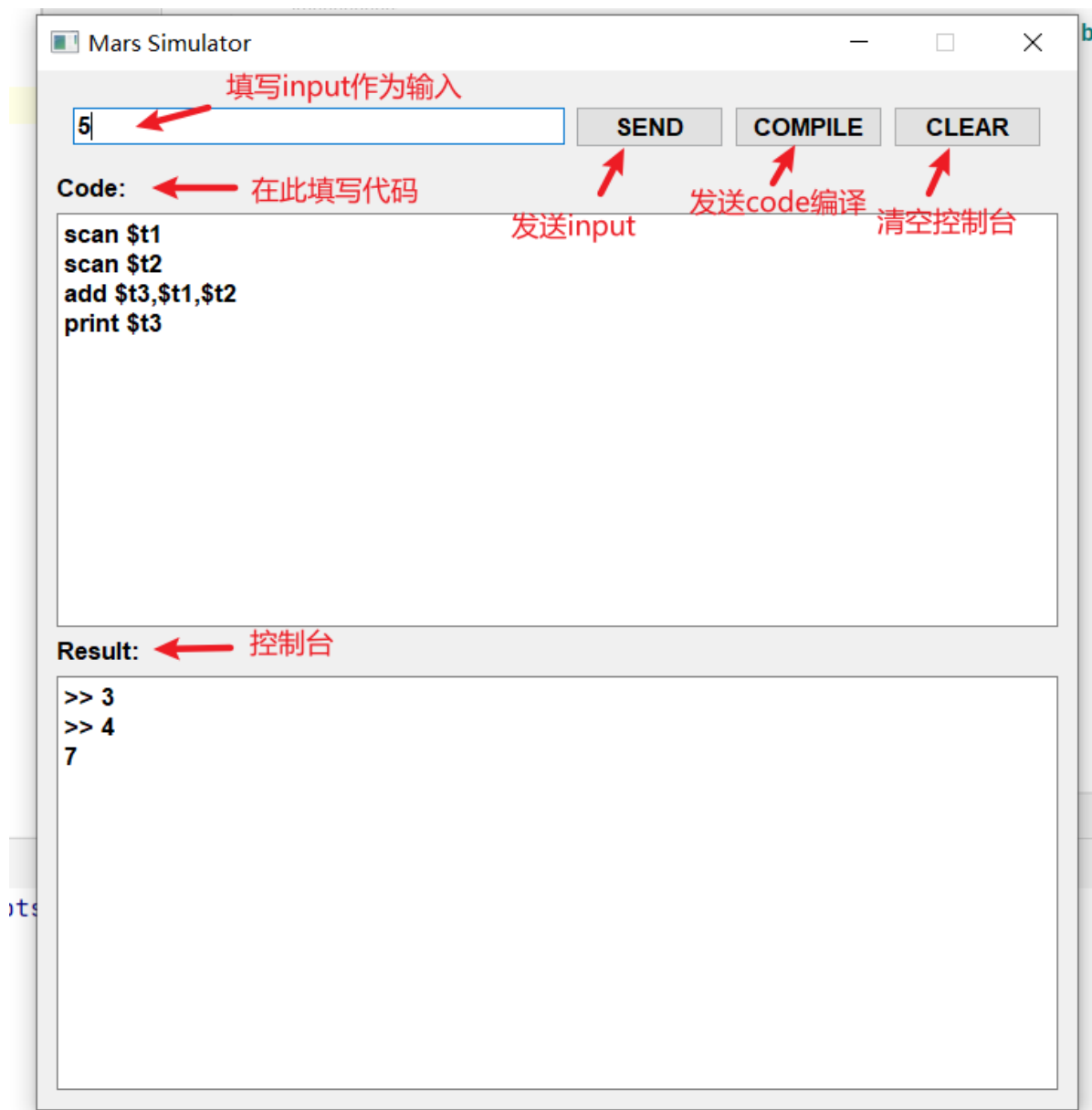
# Mars Simulator使用说明

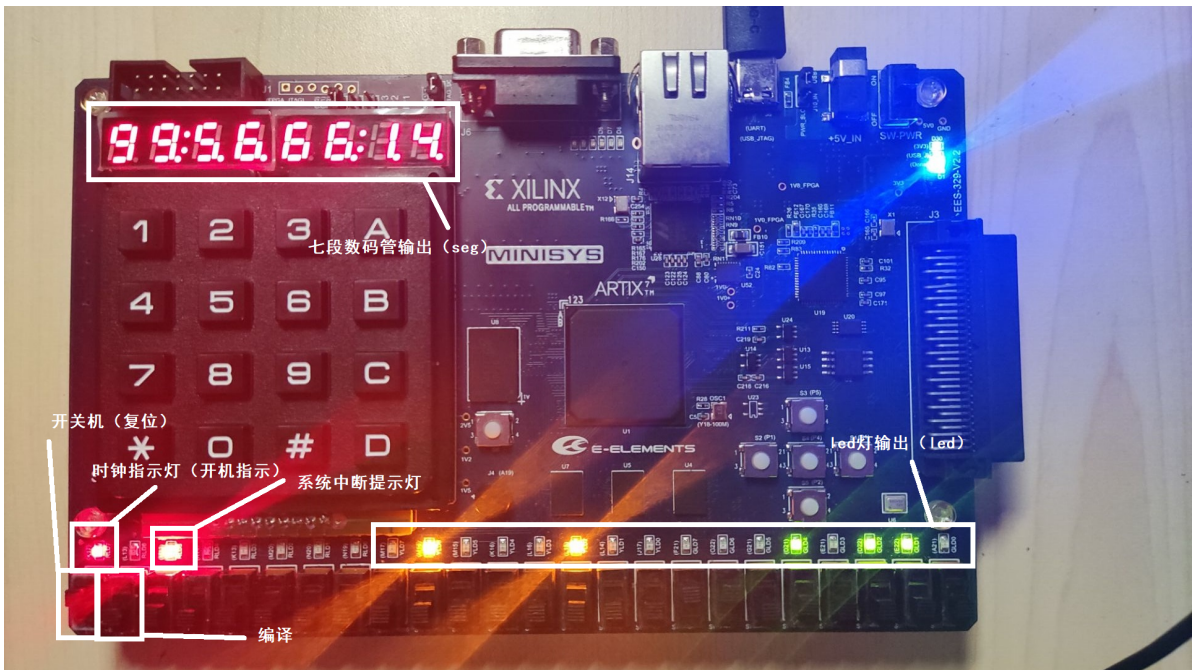
## 1 运行

将开发板与电脑连接并烧写程序；

将mips.py和main.py放在同一目录下打开main.py即可运行。运行需要一系列环境支持，建议使用pycharm打开安装环境。

## 2 使用





使用前确保开发板已与电脑连接并完成代码烧写。

只打开开发板第二个开关，启用编译。在程序端编写mips代码，点击COMPILE按钮，能够观察到系统中断提示灯熄灭（或瞬间亮一下）即为编译完成。

关闭第二个开关，打开第一个开关，即开始运行程序。需要再次运行程序时，重启第一个开关即可。需要重新编译时，关闭第一个开关，再打开第二个开关即可。

### 3 指令

本程序使用了fengyong包来进行汇编，其支持的正常使用的mips指令如下：

add	sub	and	or	jr
nor	slt	sltu	slti	sltiu
addi	lw	sw	beq	bne
j	jal			

指令编写不支持标签语法和注释，且仅能包括代码段。例如，使用bne指令时，第三个参数应填写与目标指令的距离。使用j指令时，参数应填写与目标指令的位置（第一条的位置为0）。

例如，以下代码：

```
.data
.text
loop:  addi $t1,$zero,1
        j loop
```

应写作：

```
addi $t1,$zero,1
j 0
```

本cpu指令集提供了四条额外指令：

**led \$? :** 在led灯上显示 \$? 寄存器的值

**seg \$? :** 在七段数码管上显示 \$? 寄存器的值

**print \$? :** 在控制台显示 \$? 寄存器的值

**scan \$? :** 通过控制台输入值到 \$? 寄存器

注：在运行scan指令时，系统中断提示灯会亮起，提示用户通过控制台输入。填写input后点击SEND即可。

## 4 示例

基础IO测试：

```
addi $t1,$zero,31
led $t1
seg $t1
print $t1
scan $t2
print $t2
```

加法器：

```
scan $t1
scan $t2
add $t3,$t1,$t2
print $t3
```

斐波那契数列：

```
addi $t5,$zero,1
scan $t1
addi $t2,$t2,1
add $t3,$t4,$t5
print $t3
add $t5,$zero,$t4
add $t4,$zero,$t3
bne $t2,$t1,-6
seg $t3
```

递归计算1+2+...+n：

```
scan $a0
addi $t3,$t3,1

jal 5
print $v0
j 20

addi $sp,$sp,-8
sw $ra, 4($sp)
sw $a0, 0($sp)
slt $t0,$a0,$t3
```

```
beq $t0,$zero,3
addi $v0,$zero,0
addi $sp,$sp,8
jr $ra
```

```
addi $a0,$a0,-1
jal 5
lw $a0,0($sp)
lw $ra,4($sp)
addi $sp,$sp,8
add $v0,$a0,$v0
jr $ra
```