

轮式移动机器人技术

王越

控制学院智能系统与控制研究所

ywang24@zju.edu.cn

Python通信

介绍

- Python是一门编程语言
- 简单 - 语法简单，代码易懂
- 易学 - 极其容易上手
- 免费、开源 - 自由地发布这个软件的拷贝、阅读它的源代码、对它做改动、把它的一部分用于新的自由软件中
- 高层语言 - 无需考虑诸如如何管理你的程序使用的内存一类的底层细节。



介绍

- 安装了Python解释器后，可以使用python命令
- 方式1:
 - 用python命令进入交互模式
 - 写一段python代码直接执行

执行以上命令后，出现如下窗口信息：

```
$ python3
Python 3.4.0 (default, Apr 11 2014, 13:05:11)
[GCC 4.8.2] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

在 python 提示符中输入以下语句，然后按回车键查看运行效果：

```
print ("Hello, Python!");
```

以上命令执行结果如下：

```
Hello, Python!
```

介绍

- 方式2:
 - 写一段python代码保存成文件
 - 用python命令解释这个文件

将如下代码拷贝至 `hello.py` 文件中:

```
print ("Hello, Python!");
```

通过以下命令执行该脚本:

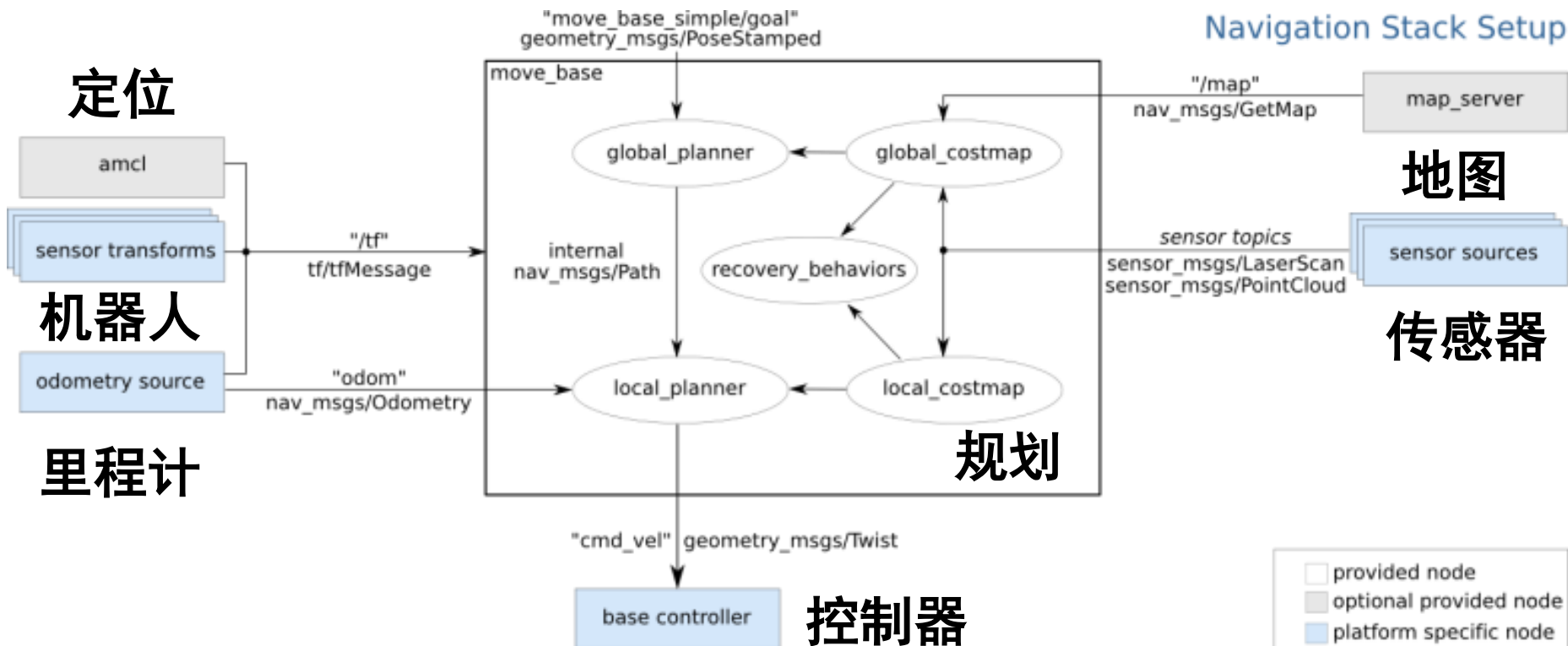
```
python3 hello.py
```

输出结果为:

```
Hello, Python!
```

- 更多资源: <http://c.biancheng.net/python/>

机器人程序



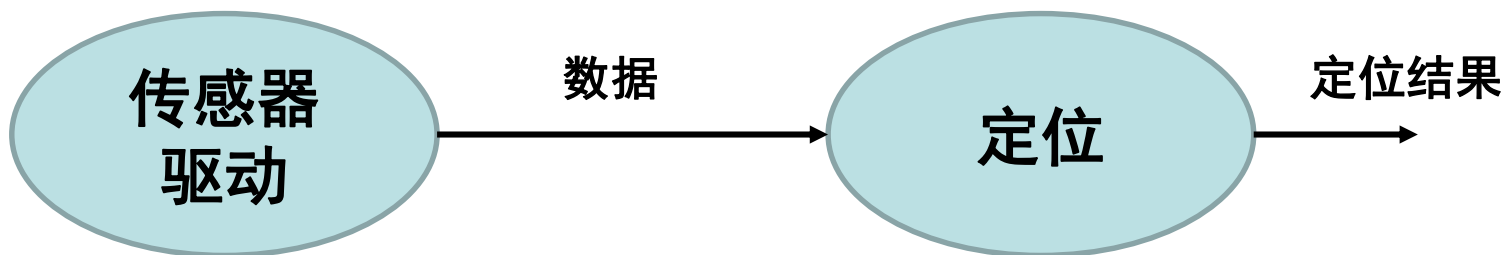
<http://wiki.ros.org/navigation/Tutorials/RobotSetup>

机器人程序

- 每个功能模块是一个进程，运行的程序
- 用python运行一个文件，对于操作系统而言运行了一个进程
- 所以定位算法就写在一个python文件，要运行定位可通过：python 定位
- 要实现多个功能模块互相协作的机器人功能，就需要：
 - 多个模块的算法代码文件，并运行
 - 多个进程之间可以互相通信

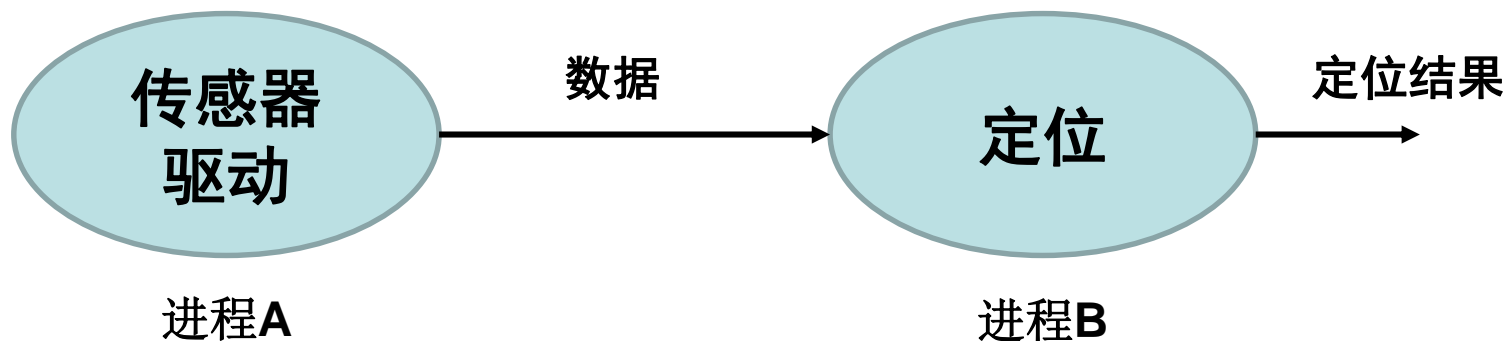
机器人程序

- 为什么要用多模块的方式对机器人编程？
 - 降低单个程序的复杂度
 - 简化调试
 - 方便重用
 - 团队开发
 - 异步运行



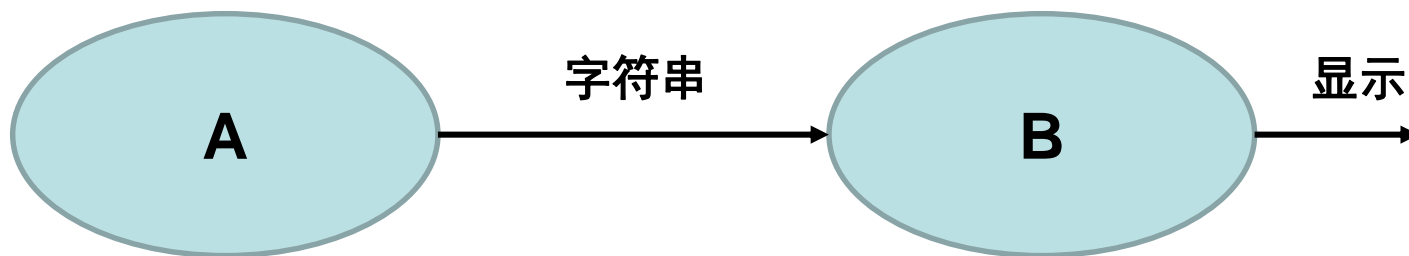
机器人程序

- 最常见的机器人程序工作方式
 - 开启进程B，等待进程A的数据
 - 开启进程A，主动产生数据，发送给进程B
 - 进程B接收到进程A的数据，执行进程B的功能
 - 进程B输出一个结果（对于用该结果的进程来说，进程B就是进程A）



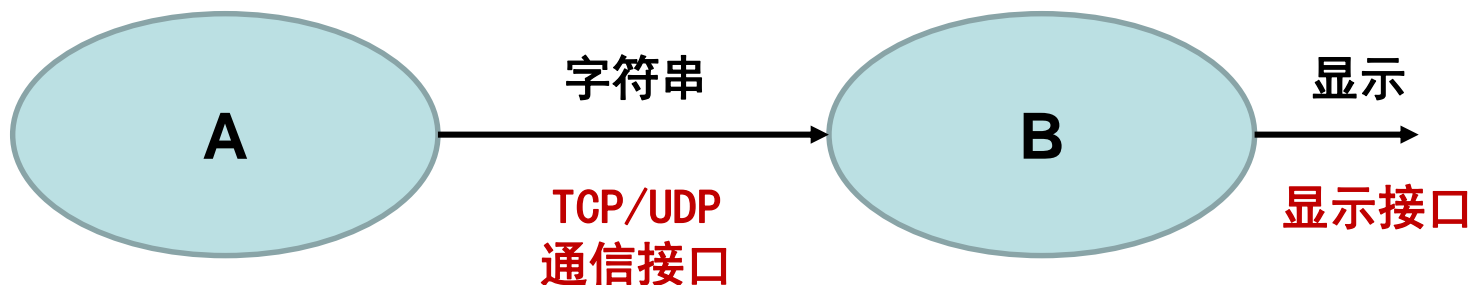
最简单案例

- 进程A发送字符串到进程B
- 进程B接受进程A的数据，并显示在屏幕上



进程间通信

- 操作系统提供了TCP/UDP通信的接口，可以用于进程间的通信，python进行了封装
- 操作系统提供了屏幕显示的接口，python进行了封装



实验作业

- 步骤1
 - 完成显示的python代码，进程B
 - 完成获取键盘并显示的python代码，进程A
- 步骤2
 - 完成进程A发送键盘内容的代码
 - 完成进程B接受键盘内容的代码
 - 启动进程A和进程B
- 需要工具：TCP/UDP通信接口，显示接口，键盘获取接口

实验作业

- 练习：
 - 进程A将某个文本文件的内容读取，发送到进程B
 - 进程B将接受到的内容进行某些编辑，并存储为另一个文件
- 需要工具：
 - 文件操作接口