



智能硬件与智能系统

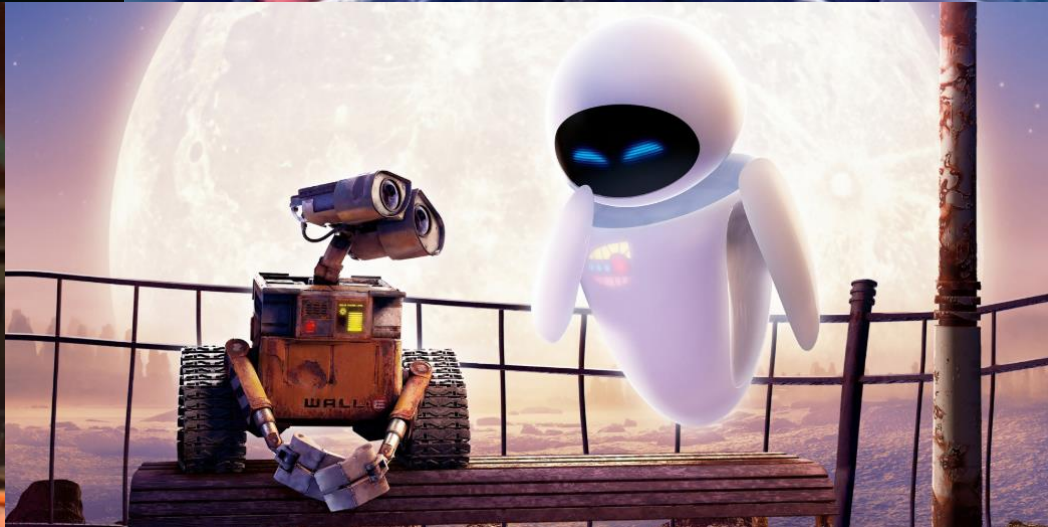
机器人操作系统ROS简介

 主讲人：刘昊天

 任课教师：陈震



那么什么是机器人？





那么什么是机器人？



工业机器人



迎宾机器人



扫地机器人



TurtleBot 2

Platforms

Boston Dynamics



SpotMini



Spot



Atlas



Handle

波士顿动力

定义

DEFINITION

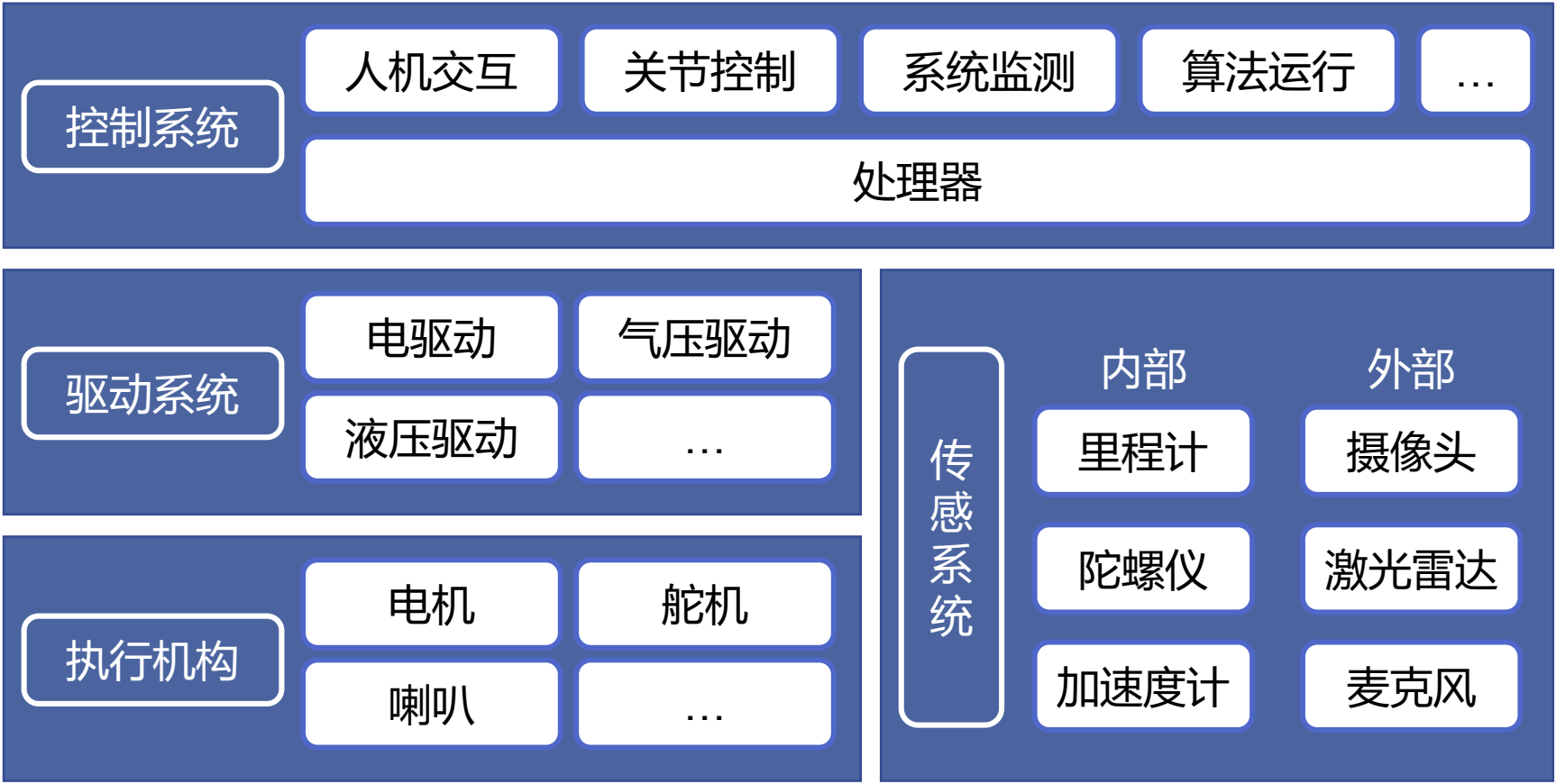
国际标准化组织(ISO)对机器人的定义如下：

- (1)机器人的动作机构具有类似于人或其他生物体的某些器官(肢体、感受等)的功能；
- (2)机器人具有通用性，工作种类多样，动作程序灵活易变；
- (3)机器人具有不同程度的智能性，如记忆、感知、推理、决策、学习等；
- (4)机器人具有独立性，完整的机器人系统中可以不依赖于人的干预。

Wikipedia: 机器人（英语：Robot）包括一切模拟人类行为或思想与模拟其他生物的机械。在当代工业中，机器人指能自动运行任务的人造机器设备，用以取代或协助人类工作，一般会机电设备，由计算机程序或是电子电路控制。



一般的机器人组成





**如何有效地组织各个部分，
形成复杂逻辑？**





ROS简介

ROS



Open Source Robotics Foundation

Standardized package manager

Runs on Ubuntu

Standardizes message-protocols for pose estimation, localization, navigation, computer vision, ...

Incorporates the latest contributions from academia

一个面向机器人的开源元操作系统。它能够提供类似传统操作系统的诸多功能，如硬件抽象、底层设备控制、进程间消息传递和程序包管理等。此外，它还提供相关工具和库，用于获取、编译、编辑代码以及在多个计算机之间运行程序完成分布式计算。ROS的运行架构是一种使用ROS通信模块实现模块间P2P的松耦合的网络连接的处理架构，它执行若干种类型的通讯，包括基于服务的同步RPC通讯、基于Topic的异步数据流通讯，还有参数服务器上的数据存储。





ROS的特点 / 为什么要用?

01

ROS支持c++、python、Java等多种开发语言。

02

底层硬件程序和高级处理程序分开，方便开发。

03

通过消息传递来协同各节点完成复杂任务，方便异步、扩展。

04

ROS 遵从BSD协议，背后有强大的开源社区支撑。

05

各种方便的测试、调试工具，如回放、日志等。

06

强迫新手们采用模块化的程序结构，而不是写成一坨。



ROS Computation Graph

- The computation graph is the network of ROS processes that are concurrently running on a robot system
- The graph consist of nodes that communicate with each other via messages
- Nodes can publish to send out messages and can subscribe to receive messages

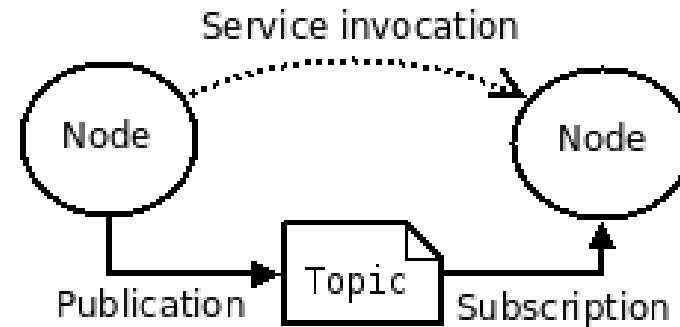
ROS Nodes

- Nodes are process that are performing some computation
- You can think of a ROS node as a running program that can read values from other nodes and also send out values



ROS Topics

- Topics are the named communication channels between nodes
- Example topics on Jet
 - /left_motor (controls the left motor)
 - /right_motor (controls the right motor)
 - /image_raw (the image from the camera)



ROS Services

- Services are procedures that answer requests.
- Services are like functions in a programming language.
- Example services on Jet
 - /data_recorder/save_image (saves the current camera image)
 - /data_recorder/start_recording (records the camera feed to a video file)



ROS Params

- Parameters are variables that store settings in ROS
- Example parameters on Jet
 - /cv_camera/image_width (width in pixels of camera image)
 - /motor_control/maxspeed (maximum speed of the motors)

Useful ROS Commands

- Commands for building source code
 - catkin_make
 - catkin_make clean
 - catkin_make install
- rostopic
 - rostopic echo
 - rostopic list
- rosservice
 - rosservice list
 - rosservice call
- roslaunch



ROS Launch

- roslaunch configures and starts multiple rosnodes
- roslaunch uses xml formatting to specify parameters for nodes
- roslaunch files are saved with the .launch extension

```
<launch>
  <node pkg="foo" type="bar" name="bar">
    <param name="param1" value="10"/>
  </node>
</launch>
```

Creating ROS Software

- Rosnodes can be written in Python or C++
- Catkin is the preferred build tool for ROS nodes.
 - all code is placed inside a catkin workspace
 - CMakeLists.txt describes the build procedure and dependencies
 - package.xml specifies dependencies and administrative information



智能硬件与智能系统

机器人操作系统ROS简介

 主讲人：刘昊天

 任课教师：陈震