A Product Oriented Design IDE for ROS Developers

ROS Conference

CANADA VANCOUVER | ☀️ 2017.9
Content

1. The pain spots of robot research and development
2. RoboWare and its value
3. Further works
Robot research and development

PAIN SPOTS
Pain spots of design and development

- Who can give me a design tool?
- Who can give me a tool to develop ROS?
- Who can help me make GUI?
- Who can help me select components?
Pain spots of design and development

- Who can give me a design tool?
- Who can give me tool to develop ROS?
- Who can help me make GUI?
- Who can help me select components?
Pain spots of design and development

01 Age of Internet
Platform: Win, Linux, Mac
Tech: Java, C#, PHP
Tool: Eclipse, VS Studio, ...

02 Age of mobile Internet
Platform: Android, iOS
Tech: Java, Obj-C
Tool: Android Studio, Xcode

03 Age of robots
Platform: Linux
Tech: ROS
Tool: None
Pain spots of design and development

- Who can give me a design tool?
- Who can give me tool to develop ROS?
- Who can help me make GUI?
- Who can help me select components?
Pain spots of design and development

- Who can give me a design tool?
- Who can give me tool to develop ROS?
- Who can help me make GUI?
- Who can help me select components?
What can help us develop robots?
Robot development IDE based on ROS

RoboWare
Make creating robots easier
RoboWare

New Concept: POD Product Oriented Design

- Designer
- Studio
- Viewer
- Robots

Develop Robots
Develop Graphic UI
The Value of Designer
What is Designer?

Drag and drop, quickly finish the hardware design.
Designer features

- Port checking
- Hardware selection
- Launch generate file
- ROS package auto downloading
We are adding some common hardware components, and your contribution will make a big difference.
The Value of Studio
What Can Studio Do?

- Create and manage ROS workspace
- Create and edit ROS codes
- Local debugging
- Remote deployment
- Remote debugging
Built-in ROS package management

Convenient installation of ROS packages/ uninstall & check Wiki

ROS package one-click-(un)install | meta-package and package searching | integrated ROS Wiki browse
# Author: Wim Meeussen

```python
from __future__ import with_statement
import rospy; roslib.load_manifest('dashgo_calibration')
import yaml
import rospy
import roslib
from sensor_msgs.msg import LaserScan
from dashgo_calibration.msg import ScanAngle
from std_msgs.msg import Int16

class ScanToAngle:
    def __init__(self):
        self.min_angle = rospy.get_param('min_angle', -0.4)
        self.max_angle = rospy.get_param('max_angle', 0.4)
        self.pub = rospy.Publisher('scan_angle', ScanAngle)
        self.sub = rospy.Subscriber('scan', LaserScan, self.scan_cb)
        self.angle_pub = rospy.Publisher('vertical_angle', Int16)

    def scan_cb(self, msg):
        angle = msg.angle_min
```

The Value of Viewer
Efficient development of CMS and UI

- Visual interface editing
- High-efficient development
- Quick response
Cross platform

Windows
Linux
Mac
iOS
Android

RoboWare Viewer Cross platform
We need you

You are welcome to provide components.
Further Works

Thinking about the future
Robo-X Projects

RoboWare
Make creating robots easier

RoboStore
Make creating robots easier

RoboSchool
Make creating robots easier
Robot components store

Chinese version is available on http://www.robostore.me
The Value of RoboStore

- Good user experience
- Precision of product description
- Label ROS compatibility
- Provide application cases solutions in different scenarios
- High quality service provider

Gathering professional hardware to support ROS
Join RoboStore

You are welcome to put your own components online to sell.
We will make it OPEN SOURCE
We have a good wish

Contribute to the community, contribute to the robot industry.
There is no end to perfection, so just keep going!

Thank you!