Principles of driver wrapping

Ingo Kresse (kresse@in.tum.de)

Intelligent Autonomous Systems Group Technische Universität München

July 18, 2011









Motivation

Example Driver

Outline

Extra Features

Conclusion

1. Motivation

2. Example Driver

3. Extra Features

4. Conclusion

July 18, 2011



Motivation Motivation

Example Driver

Extra Features

Conclusion

Why hardware drivers in ROS?

- abstract away details of the hardware
- common interface for a type of hardware
- share a piece of hardware between programs
- deal with error conditions

Ingo Kresse

July 18, 2011

Examples for drivers

Extra Features

Conclusion

ROSIEs hardware:

- ► 47 motors (arms, hands, wheels, ...)
- ► 11 sensors (lasers, cameras, kinect, ...)

They all can fail independently.



July 18, 2011

Example Driver

Extra Features

Conclusion

Functionality of a ROS drivers

- a good ROS driver should:
 - provide standard ROS topics for sensor data / commands
 - read its configuration data from roscore (rosparam)
 - restart when necessary
 - report current hardware state (rosout, diagnostics)

Driver Status			ole Dashboari	
Driver Status		Disercette	stic Power	Runstop
Driver Status	Errors			
	😄 (Actuators :	Error		
	(Actuation)	hm Left : Ox		
Full name: /Lasers/Laser Front/Driver Status	Actuators/	Arm Bight : Dx		
Component: Driver Status	Warnings			
Hardware ID: H0506272				_
Level: OK	(Actuation)	Jase : Down		
Message: OK; Streaming				
Driver state: RUNNING				
Latest status message: Streaming data.	41	100000-0000-0000-000		
Device Status Sensor works well.				_
Ports/dev/sensors/hokuyo_H0506272	7 😅 (6: 2, W: 1) /			
Device ID: H0506272	🖨 Arm Left			
Scan Thread Lost Count: 0	🖨 Arm Righ			
Corrupted Scan Count: 0	😝 Base : Do			
Vender Name: Hokzyo Automatic Co.,Ltd.		ft and right hand ol		
Product Name: SOKUIN Sensor URG-04LX Firmware Version: 3.3.00116/tor/20161	🚽 PTU Addie	in Server : We are sti	Lalive	
Pirmware Version: 3.3.00(16/Apr/2018) Protocol Version: SCIP 2.0	🚽 PTU and	Filting Laser : We are	stillalive	
Computed Latency: 90024250	▷	OK		
User Time Offset 0	P √ Lasers : OK			
1	P 🚽 Wimste: O			
1	Roac			8
				- 15
ssage				
Fixed Frame: /map				P
iubscribed to Topics: laser_front_left laser_rear_	right			1
Connected to device with ID: H0803340				
itarting calibration. This will take up a few secon				
Calibration finished. Latency is: -0.0311	Last message ree	cerved o second	s ago	
itreaming data. Connected to device with ID: H0506272				1000
Connected to device with ID: H0506272 Connected to device with ID: H0506270		1000		100
Santiected to device with ID: H0500270 Starting calibration. This will take up a few second	4	Info		ser front
scarting calibration. This will take up a few secon Starting calibration. This will take up a few secon		Info		Ger_rear
atting footprint, x : 0.4300000.530000 v: 4		Info	1	peed filter
row've passed in true for auto start for the C++ a				av ocontro
utting footprint.x:-0.4300000.530000 y:-1		into		av_pcontro
rou've passed in true for auto_start for the C++ a		ass in false War		aw_pcontro
Calibration Einisted, Latency is -0.0960		info		sec front
Calibration finished. Latency is: -0.1000		1000		Sec.rear
Streaming data.		1470		Sec.rear
Streaming data.		Info		ser front
o taring out.				and the second

Ingo Kresse

July 18, 2011

Motivation	Exa
	Outline

Conclusion

1. Motivation

2. Example Driver

3. Extra Features

4. Conclusion

July 18, 2011



Interface design

Extra Features

Conclusion

Things to consider:

- Hardware capabilities:
 - What commands does the driver understand?
 - What data does the driver publish?
- What messages exist that convey this information? (> rosmsg)
- How are other drivers doing it?

July 18, 2011





Conclusion

Example: mobile robot base

Interface

commanded velocity: geometry_msgs/Twist

odometry: tf

Advantages of this interface:

- all based on topics (can be monitored, recorded, etc.)
- used by other drivers as well
- simple

July 18, 2011



Example Driver

Extra Features

Conclusion

Ingo Kresse

Starting to code

Let's use roscpp:

manifest.xml:

```
...
<depend package="roscpp"/>
<depend package="geometry_msgs"/>
<depend package="tf"/>
...
```

src/cool_new_base.cpp:

```
#include <ros/ros.h>
...
int main(int argc, char *argv[])
{
    ros::init(argc, argv, "cool_new_base");
    ros::NodeHandle nh("~");
    ...
}
```

July 18, 2011

Example Driver

Extra Features

Conclusion

Ingo Kresse

Receiving commands

Let's parse the twist command:

```
src/cool_new_base.cpp:
#include <geometry_msgs/Twist.h>
. . .
void callback(const geometry_msgs::Twist::ConstPtr& msg)
    double fwd = msg->linear.x;
    double rot = msg->angular.z;
    . . .
main(...)
    . . .
    ros::Subscriber sub = nh.subscribe("/cmd_vel", 1, callback);
```

July 18, 2011



Conclusion

Ingo Kresse

We provide the odometry by publishing a tf transform:

```
src/cool_new_base.cpp:
#include <tf/transform_broadcaster.h>
. . .
    tf::TransformBroadcaster transforms;
    . . .
    tf::Transform pose(tf::Quaternion(tf::Vector3(0, 0, 1), theta),
                       tf::Point(x, y, 0.0));
    transforms.sendTransform(tf::StampedTransform(pose, ros::Time::now(),
                              "/odom", "/base_link"));
    . .
```

July 18, 2011



Give ROS a chance to work:

```
src/cool_new_base.cpp:
. . .
while(true) // main loop
{
    . . .
    ros::spinOnce();
}
```

This makes shure that messages are handled and callbacks are called.

Ingo Kresse

July 18, 2011



How to organize the main loop:

- loop rate required by the device: Usually there is a 'wait-for-new-data' anyway.
 - \rightarrow publish every nth data sample.
- (more or less) arbitrary loop rate: Use ros::Rate.
- Avoid busy waiting (uses 100% CPU)

Conclusion



Some more things to consider:

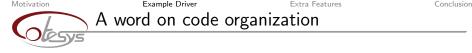
- Callbacks should not block for a long time (everything else will be stalled as well)
- Keep computational load within bounds (also for the consumers)
- Set the data rate 'right': (rule of thumb in ROS: 10-200 Hz. Best option: have a rosparam!)

Conclusion

Ingo Kresse

Make 'good' time stamps (calibrate delays, if possible)

July 18, 2011



What about the rest of the code?

- Encapsulate hardware code into separate class (separate file)
- Make separate ROS wrapper class.

Advantages:

- hardware code is reusable in other frameworks
- ▶ ROS wrapper has an easier job to e.g. restart.

July 18, 2011



Example Driver

Extra Features

Conclusion

Robustness

How can we make a driver robust?

- check whether data is arriving
- check whether data is valid
- when something is wrong: reset device
- keep trying!

July 18, 2011



	atio	

Example Driver

Outline

Extra Features

Conclusion



2. Example Driver

3. Extra Features

4. Conclusion

July 18, 2011

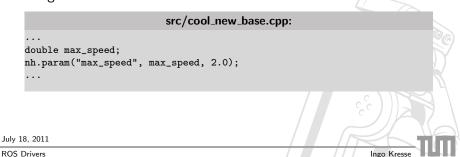


Motivation	Example Driver
	rosparams

ROS Parameters

- do the job of reading parameters from file or command line
- stored in a central place (roscore)
- usually primitive types, but can be structs and lists
- there are tools to change them online

Usage:



Ingo Kresse



How can we set a ROS Parameter?

When starting from the command line:

command mie.	command	line:
--------------	---------	-------

- # ./cool_new_base _max_speed:=0.5
- Using the rosparam tool:

rosparam tool:

rosparam set cool_new_base/max_speed 0.5

In a launch file:

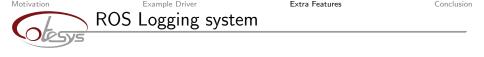
. . .

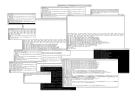
. . .

cool_new_base.launch

```
<param name="max_speed" type="double" value="0.5" />
```

July 18, 2011





VS.

Ingo Kresse

- has severity levels (DEBUG, INFO, WARN, ERROR)
- reports file name and line number
- travels over the network

July 18, 2011

Example Driver

Extra Features

Conclusion

ROS Logging system

powerful console

1410	nsole		_ •
Message	Severity	Node	Time
β hello world 78856	Info	Aaker	1259701549.969195000
🚯 hello world 78867	Info	/talker	1259701550.069209000
hello world 78050	Info	/taloer	1259701550.169192000
🕼 helio world 78872	Info	/taker	1259701550.269198000
🔋 helio world 78873	Info	/talker	1259701550.369194000
🐌 helio world 78874	Info	/talker	1259701550,469195000
🕼 helo world 78875	Info	/talcer	1259701550.569196000
🔋 helio world 78876	Info	/taker	1259701550.669191000
🗊 helio world 78677	Info	/talker	1259701550.769193000
🕼 helio world 78878	Info	/talcer	1259701550.869224000
🔋 helio world 78879	Info	/taker	1259701550.969351000
🗊 helio world 78680	Info	/talker	1259701551.069208000
🐌 helio world 70001	Info	/talcer	1259701551.169190000
🔋 helio world 78892	Info	/taker	1259701551.269193000
🗊 helio world 78883	Info	/talker	1259701551.369193000
🚺 helio world 70004	Info	/talcer	1259701551.469194000
🔋 helio world 78895	Info	/taker	1259701551.569194000
🔋 helio world 78886	Info	/talker	1259701551.669190000
🐌 helio world 78887	Info	/talker	1259701551.769207000
helio world 78898	Info	Aaker	1259701551,869196000
🔋 helio world 78889	Info	/taker	1259701551.969193000
🗊 helio world 78890	Info	/talker	1259701552.069209000
🛿 helio world 78891	Info	Aalcer	1259701552.169190000
🔋 helio world 78892	Info	/taker	1259701552.269193000
bello world 78893	Info	/talker	1259701552.369192000
cieverity 🖸 Fatal 👩 Error 🕲 Warn 🕲 Info 🕲 Debug		Pause Clear	Setup Levels New Windo
2 Enabled	Include 0 Pegex	From 🖾 Message 🖸 Nod	e 🔁 Location 😰 Topics 😂 🌷

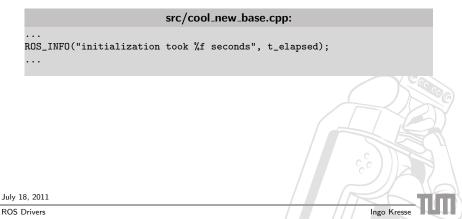
- all message in one place
- extensive filtering capabilities

July 18, 2011



Conclusion

Usage example:



Diagnostics

11	Driver Status		- I- IX
Full pages () accord	Laser Front/Driver Status		
Component: Driver			
Hardware ID: H050			
evel: CK			
Message: OK; Strea	ming		
Driver state: RUNI	NING		
Latest status mess	age: Streaming data.		
Device Status: Sen	sor works well.		
Port: /dev/sensors/	hokuyo_H0506272		
Device ID: H05062	72		
Scan Thread Lost 0			
Corrupted Scan Co			
	uyo Automatic Co.,Ltd.		
	KUIKI Sensor URG-04LX		
	3.3.00(16/Apt/2008)		
Protocol Version:			
Computed Latency			
User Time Offset:	2		
	0	Pause	Snapshot



Purpose:

- keep track of hardware status
- quickly find out "why it's not working"

Regularly (every 1s) send some status message, containing drivers state.

July 18, 2011



Example Driver

Diagnostics

Extra Features

Conclusion

Ingo Kresse

Code example:

```
src/cool_new_base.cpp:
void diagsUpdate(diagnostic_updater::DiagnosticStatusWrapper &s)
 s.summary(diagnostic_msgs::DiagnosticStatus::OK, "Operational");
  s.addf("important number", "%d", 42);
  . . .
 diagnostic_updater::Updater diags;
 diags.setHardwareID("cool_new_base");
 diags.add("Base", diagsUpdate);
  . . .
 diags.update();
```

July 18, 2011

	ation	

Example Driver

Extra Features

Conclusion

Dynamic reconfigure

Reconfigure	
Eile	
/camera_synchronizer_node	~
projector_rate: 40.0 120.0 58.82	3529
projector_pulse_length: 0.001 0.002 0.002	
projector_pulse_shift: 0.0 1.0	-1
projector_mode: ProjectorAuto (2)	~
prosilica_projector_disable: 🗹	
stereo_rate: 1.0	1764
wide_stereo_trig_mode: WithoutProjector (4)	~
narrow_stereo_trig_mode: WithoutProjector (4)	~
forearm_r_rate: 1.0 00.0 30.0	
forearm_r_trig_mode: InternalTrigger (1)	~
forearm_l_rate: 1.0 00.0 30.0	
forearm_l_trig_mode: InternalTrigger (1)	~
projector_tweak: -0.1 0.0	
camera_reset:	

- Drivers publish details about their parameters (min/max values, etc.) on a topic.
- From this description a GUI is generated.
- The changed configuration is sent using a service call.

July 18, 2011



	/atio	

Conclusion



- 1. Motivation
- 2. Example Driver
- 3. Extra Features
- 4. Conclusion

July 18, 2011



Well, every presentation should have one, right?

- Philosophy is geared towards systems with a lot of hardware

 → robustness is a must
- ROS code interface is easy (only a few lines of code)
- There are many ROS tools that help with common tasks (rosparams, rosout, diagnostics, etc.)

July 18, 2011

