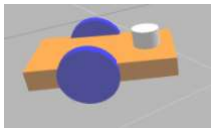
Robotique

PARTIE 2

EL 3007

R. HAMOUCHE
r.hamouche@esiee.fr

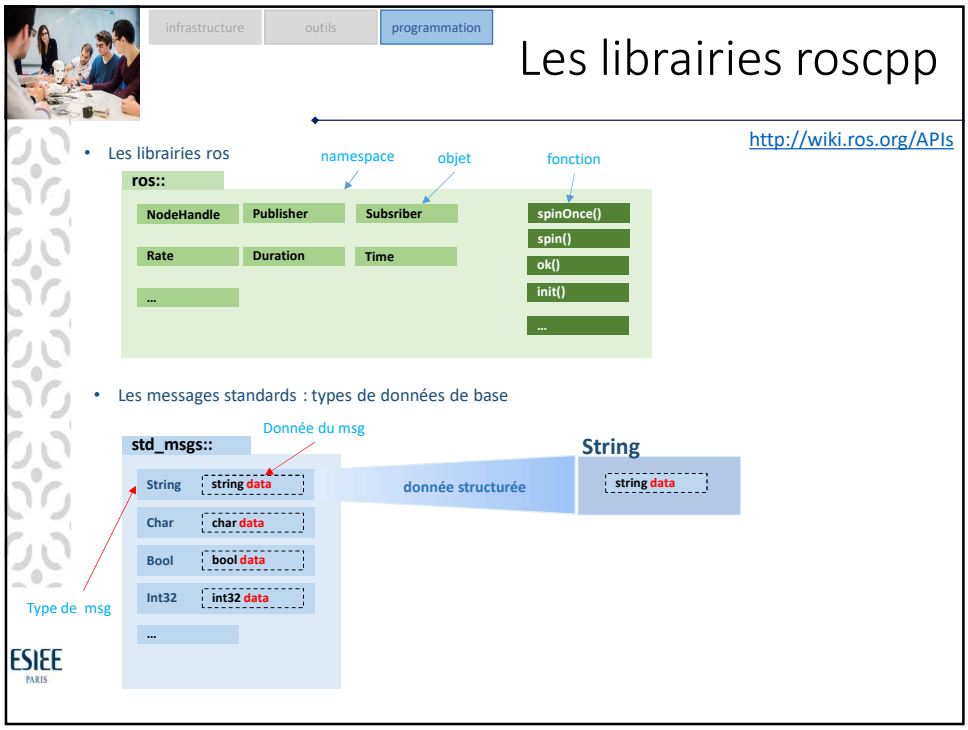


2023-2024

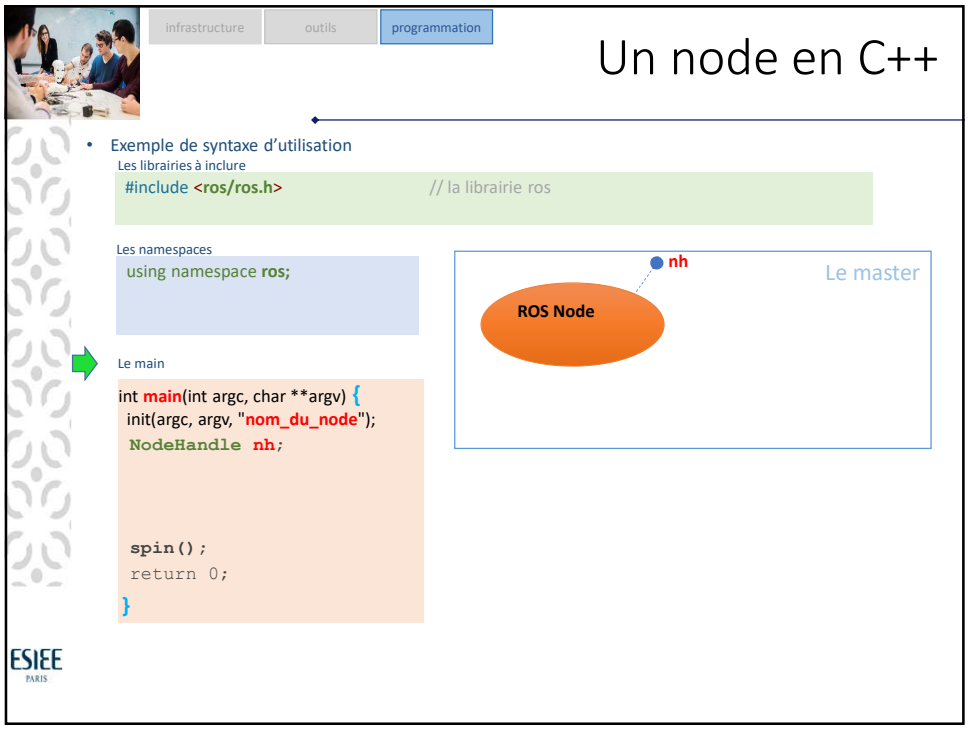
1

CRÉER DES APPLICATIONS ROS

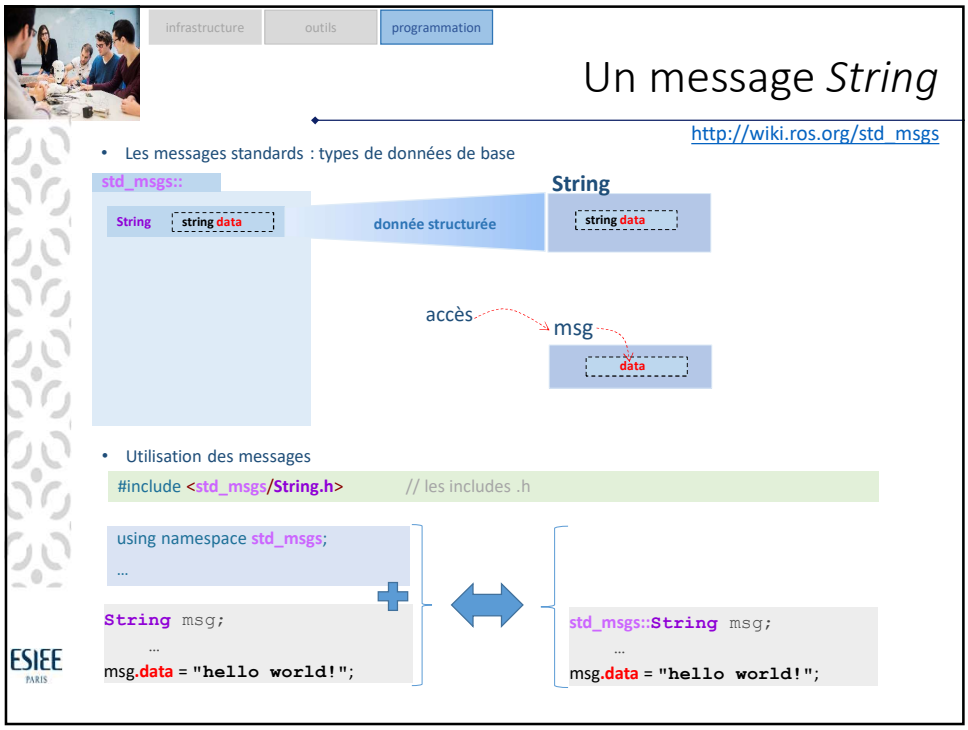
2


infrastructure
outils
programmation
Les librairies roscpp
http://wiki.ros.org/APIs
Les librairies ros
namespace
objet
fonction
ros::
NodeHandle
Publisher
Subsriber
Rate
Duration
Time
...
spinOnce()
spin()
ok()
init()
...
Les messages standards : types de données de base
Donnée du msg
std_msgs::
String
string data
Char
char data
Bool
bool data
Int32
int32 data
...
donnée structurée
String
string data
Type de msg
ESIEE
PARIS

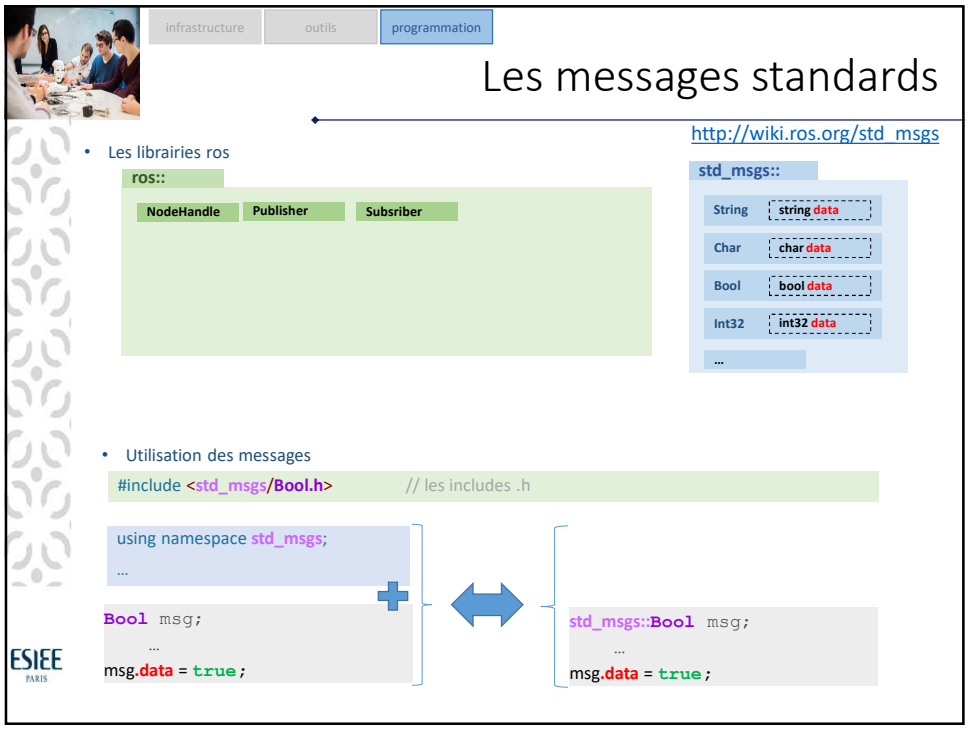
3


infrastructure
outils
programmation
Un node en C++
Exemple de syntaxe d'utilisation
Les librairies à inclure
#include <ros/ros.h>
// la librairie ros
Les namespaces
using namespace ros;
Le main
int main(int argc, char **argv) {
init(argc, argv, "nom_du_node");
NodeHandle nh;
spin();
return 0;
}
nh
ROS Node
Le master
ESIEE
PARIS

4


infrastructure
outils
programmation
Un message String
http://wiki.ros.org/std_msgs
Les messages standards : types de données de base
std_msgs::
String
string data
donnée structurée
String
string data
accès
msg
data
Utilisation des messages
#include <std_msgs/String.h>
// les includes .h
using namespace std_msgs;
...
String msg;
...
msg.data = "hello world!";
std_msgs::String msg;
...
msg.data = "hello world!";
ESIEE
PARIS

5


infrastructure
outils
programmation
Les messages standards
http://wiki.ros.org/std_msgs
Les librairies ros
ros::
NodeHandle
Publisher
Subsriber
std_msgs::
String
string data
Char
char data
Bool
bool data
Int32
int32 data
...
Utilisation des messages
#include <std_msgs/Bool.h>
// les includes .h
using namespace std_msgs;
...
Bool msg;
...
msg.data = true;
std_msgs::Bool msg;
...
msg.data = true;
ESIEE
PARIS

6

infrastructure | outils | programmation

Un publisher en C++

- Exemple de syntaxe d'utilisation

Les librairies à inclure

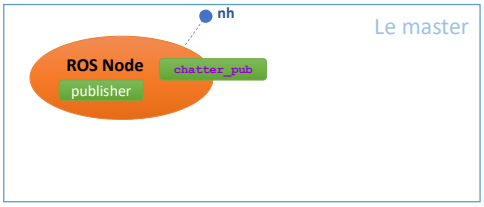
```
#include <ros/ros.h>                // la librairie ros
#include <std_msgs/String.h>        // le type string pour les messages
```

Les namespaces

```
using namespace ros;
using namespace std_msgs;
```

La déclaration des variables

```
NodeHandle nh;
String msg;
Publisher chatter_pub;
```



ESIEE PARIS

7

infrastructure | outils | programmation

Un publisher en C++

- Exemple de syntaxe d'utilisation

Les librairies à inclure

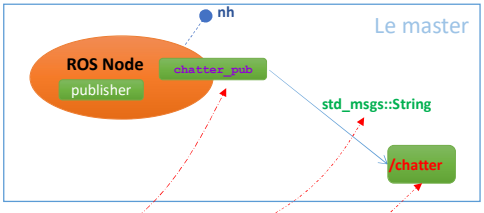
```
#include <ros/ros.h>                // la librairie ros
#include <std_msgs/String.h>        // le type string pour les messages
```

Les namespaces

```
using namespace ros;
using namespace std_msgs;
```

La déclaration des variables

```
NodeHandle nh;
String msg;
Publisher chatter_pub;
```



Dans le main :

La création du publisher vers un topic

```
chatter_pub = nh.advertise<String>("/chatter", 10);
```

Publieur de msg sur un topic donné

Type de msg

Nom du topic

ESIEE PARIS

8

infrastructure | outils | programmation

Un publisher en C++

- Exemple de syntaxe d'utilisation

Les librairies à inclure

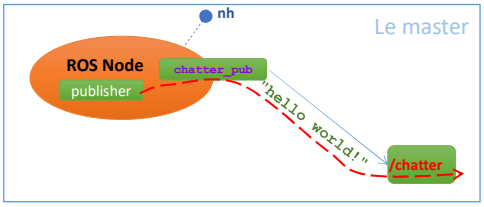
```
#include <ros/ros.h>            // la librairie ros
#include <std_msgs/String.h>    // le type string pour les messages
```

Les namespaces

```
using namespace ros;
using namespace std_msgs;
```

La déclaration des variables

```
NodeHandle nh;
String msg;
Publisher chatter_pub;
```



Dans le main :

La création du publisher vers un topic

```
chatter_pub = nh.advertise<String>("/chatter", 10);
Duration(1).sleep();
```

L'envoi des messages

```
msg.data = "hello world!";
chatter_pub.publish(msg);
```

msg

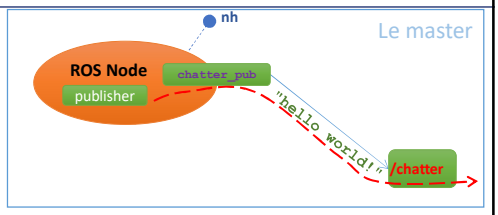
```
data="hello world!";
```

ESIEE PARIS

9

infrastructure | outils | programmation

Un publisher périodique

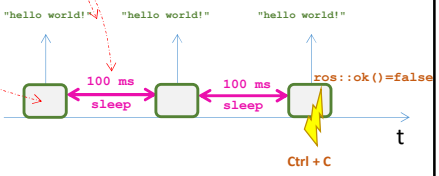


Dans le main :

- L'envoi périodique des messages

```
Rate loop_rate(10);
while(ok()){
loop_rate.sleep();
msg.data = "hello world!";
chatter_pub.publish(msg);
spinOnce();
}
```

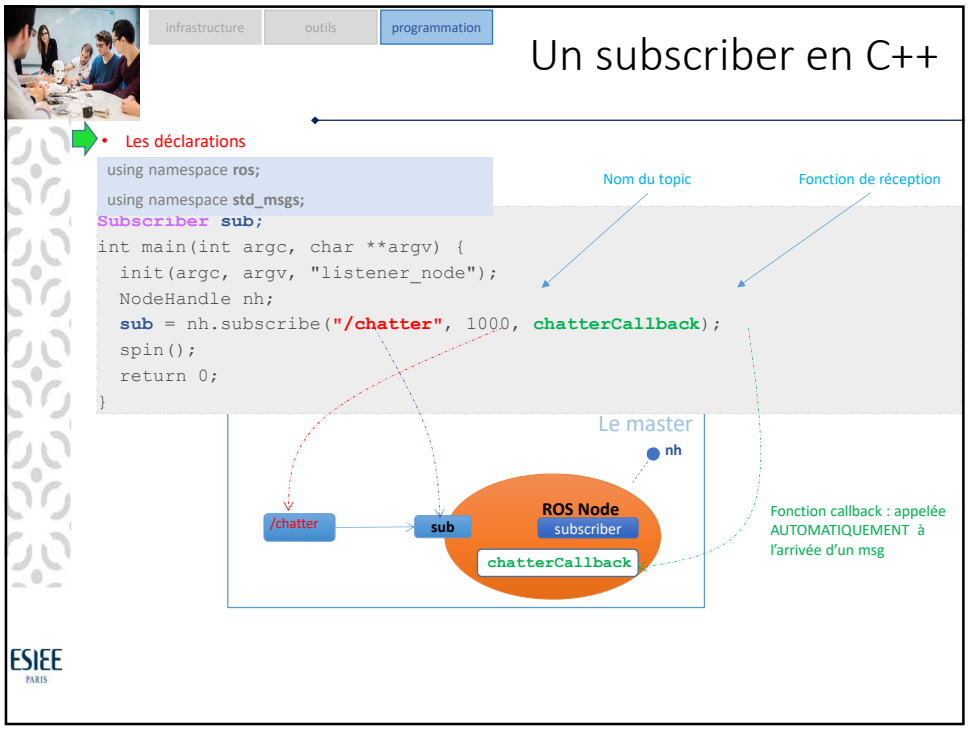
Fréquence 10hz → 100 ms



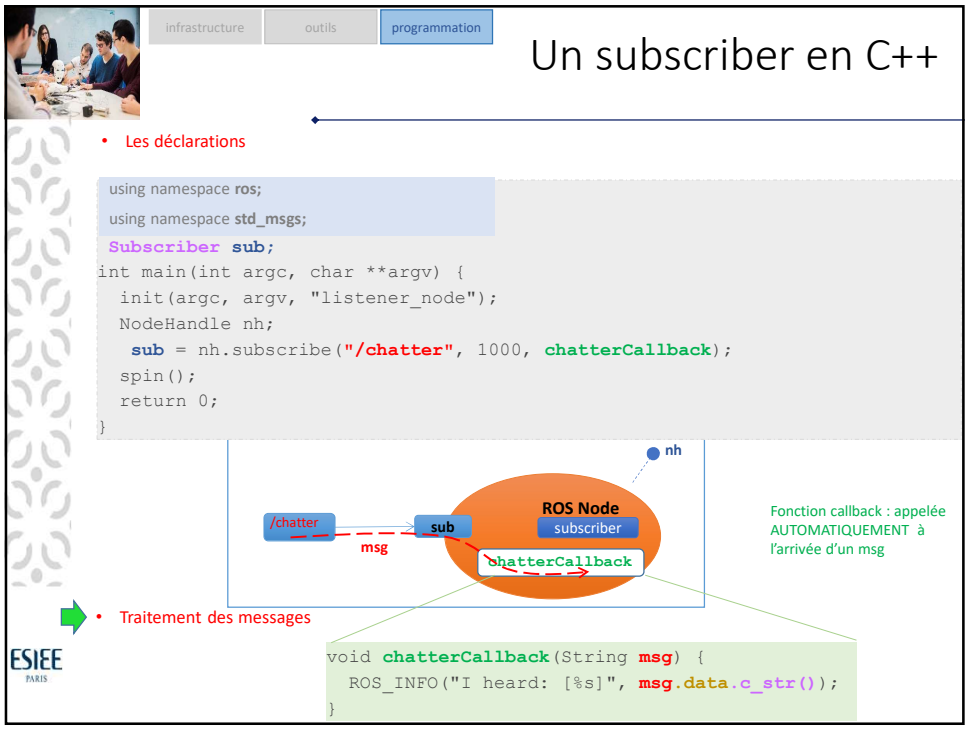
Permet de traiter les messages ROS en attente

ESIEE PARIS

10


infrastructure
outils
programmation
Un subscriber en C++
Les déclarations
using namespace ros;
using namespace std_msgs;
Subscriber sub;
int main(int argc, char **argv) {
init(argc, argv, "listener_node");
NodeHandle nh;
sub = nh.subscribe("/chatter", 1000, chatterCallback);
spin();
return 0;
}
Nom du topic
Fonction de réception
Le master
nh
/chatter
sub
ROS Node
subscriber
chatterCallback
Fonction callback : appelée AUTOMATIQUEMENT à l'arrivée d'un msg
ESIEE
PARIS

11


infrastructure
outils
programmation
Un subscriber en C++
Les déclarations
using namespace ros;
using namespace std_msgs;
Subscriber sub;
int main(int argc, char **argv) {
init(argc, argv, "listener_node");
NodeHandle nh;
sub = nh.subscribe("/chatter", 1000, chatterCallback);
spin();
return 0;
}
nh
/chatter
sub
msg
ROS Node
subscriber
chatterCallback
Fonction callback : appelée AUTOMATIQUEMENT à l'arrivée d'un msg
Traitement des messages
void chatterCallback(String msg) {
ROS_INFO("I heard: [%s]", msg.data.c_str());
}
ESIEE
PARIS

12

ROS time

using namespace **ros;**

- **Fréquence**
  - Rate rate(10); // 10Hz
- **Temps**
  - Time debut = Time::now(); // l'instant courant
- **Durée**
  - Duration duree(2); // 2s
  - Duration(2).sleep(); // suspend l'execution du node pendant 2sec
  - Duration duree_ecoulee = Time::now() - debut; //durée écoulée

ESIEE PARIS

13

infrastructure | outils | programmation

Les messages communs

http://wiki.ros.org/common_msgs

- Les librairies ros

std_msgs::

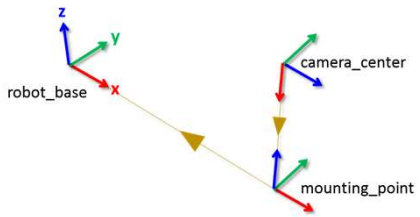
String	string data
Char	char data
Bool	bool data
Int32	int32 data
...	

geometry_msgs::

- Vector3 → Position du robot, capteur…
- Pose2D
- Twist
- Quaternion
- ...

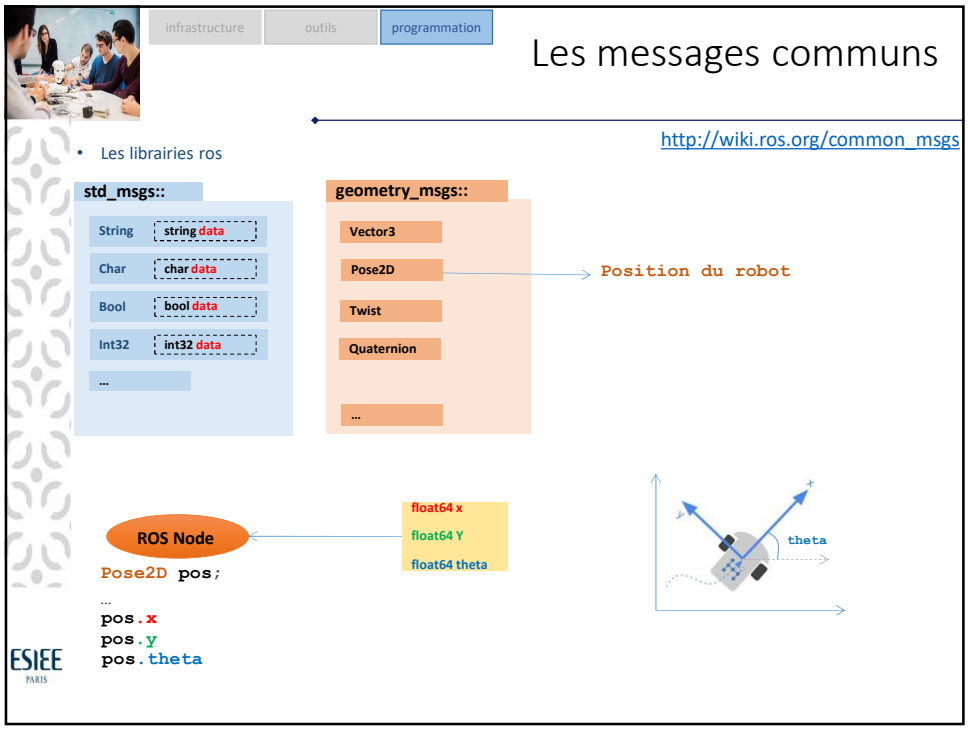
ROS Node ← float64 x, float64 Y, float64 z

```
Vector3 pos;
...
pos.x
pos.y
pos.z
```

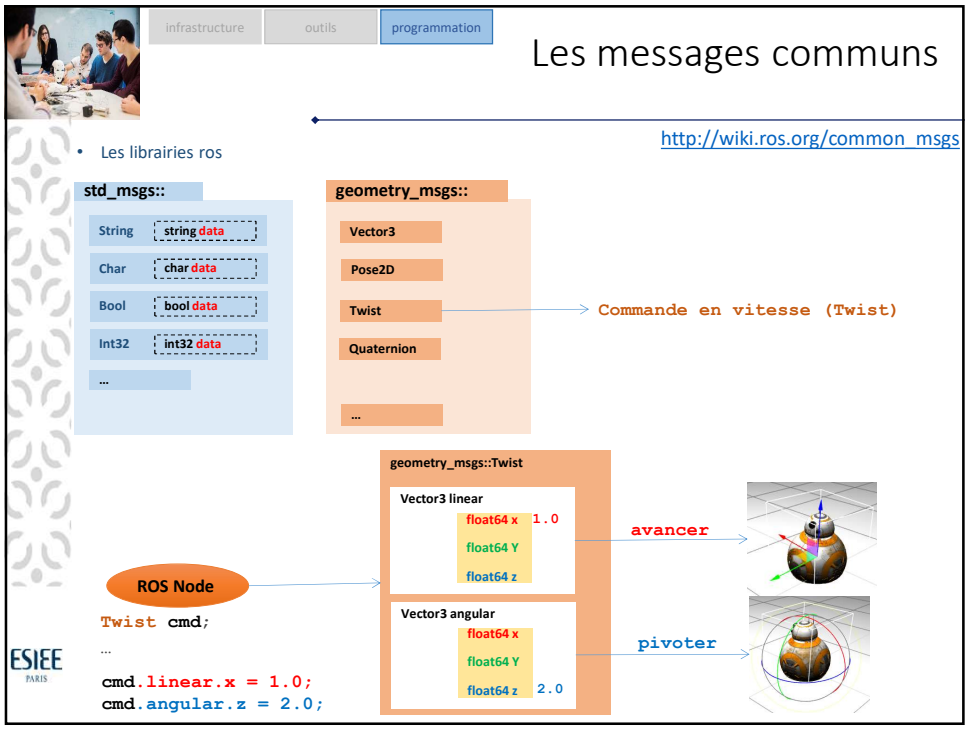


ESIEE PARIS

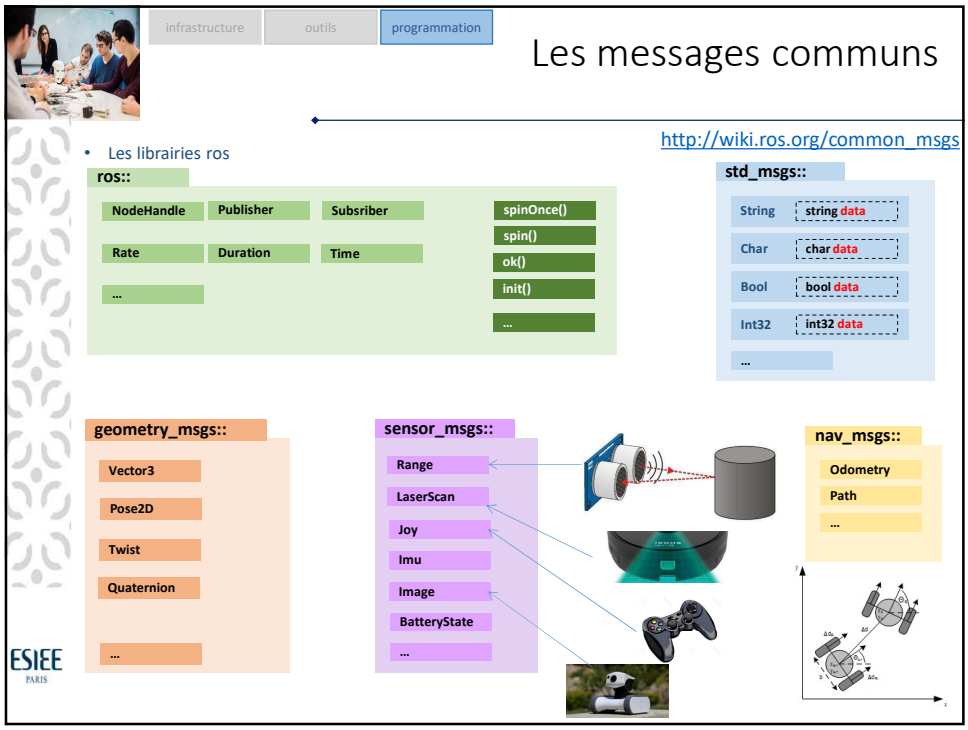
14


infrastructure
outils
programmation
Les messages communs
http://wiki.ros.org/common_msgs
Les librairies ros
std_msgs::
String
string data
Char
char data
Bool
bool data
Int32
int32 data
...
geometry_msgs::
Vector3
Pose2D
Twist
Quaternion
...
Position du robot
ROS Node
float64 x
float64 Y
float64 theta
Pose2D pos;
...
pos.x
pos.y
pos.theta
theta
ESIEE
PARIS

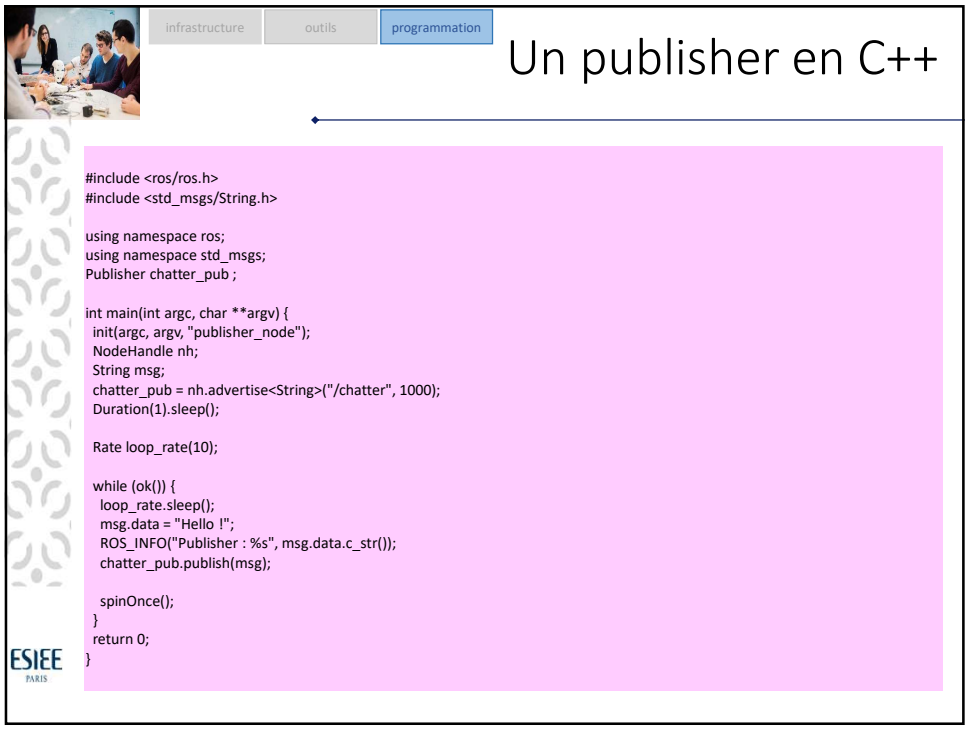
15


infrastructure
outils
programmation
Les messages communs
http://wiki.ros.org/common_msgs
Les librairies ros
std_msgs::
String
string data
Char
char data
Bool
bool data
Int32
int32 data
...
geometry_msgs::
Vector3
Pose2D
Twist
Quaternion
...
Commande en vitesse (Twist)
geometry_msgs::Twist
Vector3 linear
float64 x
1.0
float64 Y
float64 z
avancer
Vector3 angular
float64 x
float64 Y
float64 z
2.0
pivoter
ROS Node
Twist cmd;
...
cmd.linear.x = 1.0;
cmd.angular.z = 2.0;
ESIEE
PARIS

16


infrastructure
outils
programmation
Les messages communs
http://wiki.ros.org/common_msgs
Les librairies ros
ros::
NodeHandle
Publisher
Subsriber
Rate
Duration
Time
...
spinOnce()
spin()
ok()
init()
...
std_msgs::
String
string data
Char
char data
Bool
bool data
Int32
int32 data
...
geometry_msgs::
Vector3
Pose2D
Twist
Quaternion
...
sensor_msgs::
Range
LaserScan
Joy
Imu
Image
BatteryState
...
nav_msgs::
Odometry
Path
...
ESIEE
PARIS

17


infrastructure
outils
programmation
Un publisher en C++
#include <ros/ros.h>
#include <std_msgs/String.h>
using namespace ros;
using namespace std_msgs;
Publisher chatter_pub ;
int main(int argc, char **argv) {
init(argc, argv, "publisher_node");
NodeHandle nh;
String msg;
chatter_pub = nh.advertise<String>("/chatter", 1000);
Duration(1).sleep();
Rate loop_rate(10);
while (ok()) {
loop_rate.sleep();
msg.data = "Hello !";
ROS_INFO("Publisher : %s", msg.data.c_str());
chatter_pub.publish(msg);
spinOnce();
}
return 0;
}
ESIEE
PARIS

18

infrastructure | outils | programmation

Un subscriber en C++

```
#include <ros/ros.h>
#include <std_msgs/String.h>

using namespace ros;
using namespace std_msgs;
Subscriber sub ;

void chatterCallback(String msg) {
 ROS_INFO("Chatter Listener : I heard: [%s]", msg.data.c_str());
}

int main(int argc, char **argv) {
  init(argc, argv, "listener_node");
  NodeHandle nh;

  sub = nh.subscribe("/chatter", 1000, chatterCallback);

  spin();

  return 0;
}
```

ESIEE PARIS

19

ROSSERIAL - ARDUINO

20

infrastructure | outils | programmation

ROSSERIAL - publisher

- Exemple de syntaxe d'utilisation

Les librairies à inclure

```
#include <ros/ros.h>            // la librairie ros
#include <std_msgs/String.h>    // le type string pour les messages
```

Les namespaces

```
using namespace ros;
using namespace std_msgs;
```

La déclaration des variables

```
NodeHandle nh;
Std_msgs::String msg;
Publisher chatter_pub("/chatter", &msg);
```

SETUP() La création du publisher vers un topic

```
nh.initNode();
nh.advertise(chatter_pub);
```

Dans SETUP ou LOOP() L'envoi des messages

```
msg.data = "hello world!";
chatter_pub.publish(&msg);
```

msg

```
data="hello world!";
```

ESIEE PARIS

21

infrastructure | outils | programmation

Un subscriber en C++

```
#include <ros/ros.h>            // la librairie ros
#include <std_msgs/String.h>    // le type string pour les messages
```

- Les déclarations

```
using namespace ros;
using namespace std_msgs;
```

- Callback : Traitement des messages

```
void chatterCallback(std_msgs::String msg) {
  // la donnee dans msg.data
}
```

```
Subscriber<std_msgs::String> chatter_sub("/chatter", &chatterCallback);
```

Dans SETUP

```
nh.initNode();
nh.subscribe(chatter_sub);
```

ESIEE PARIS

22