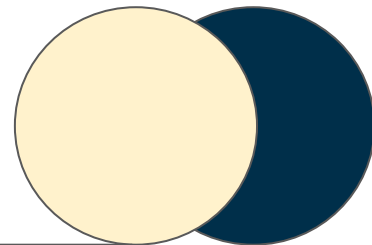


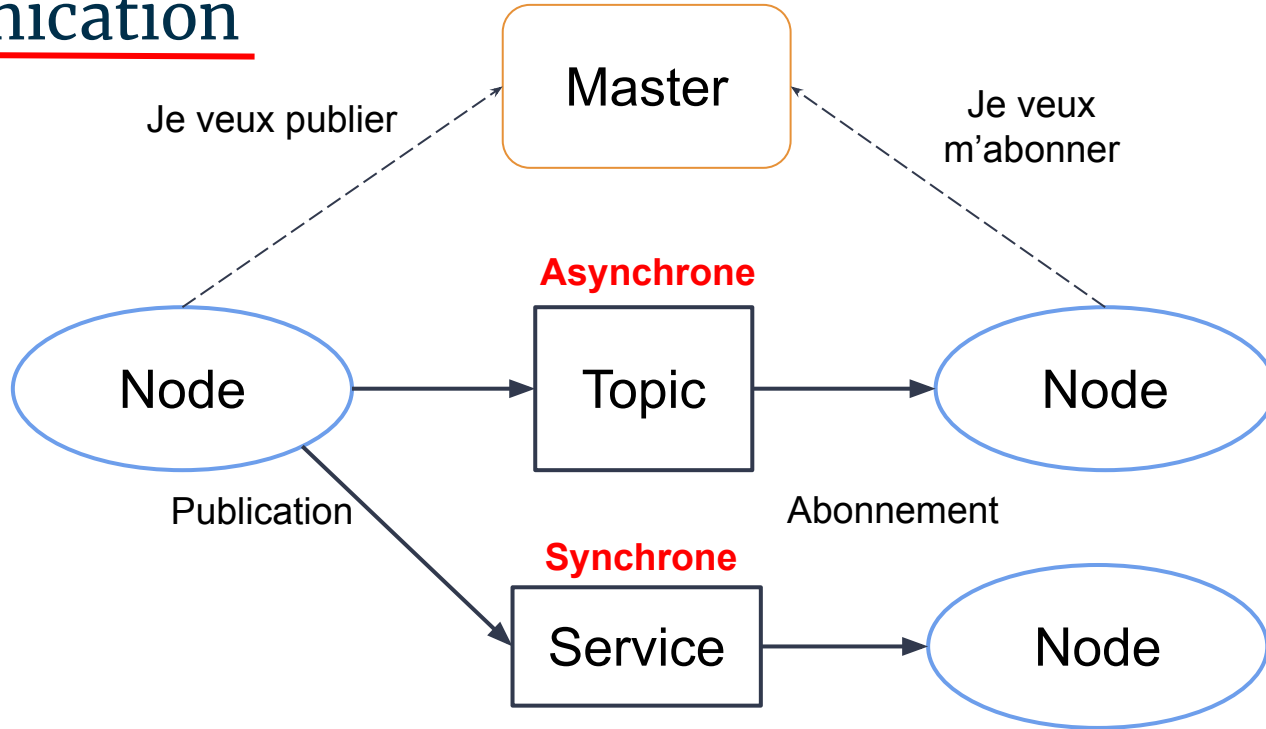
---

BONUS





## Communication



# Protocole Simple Message

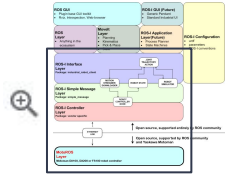
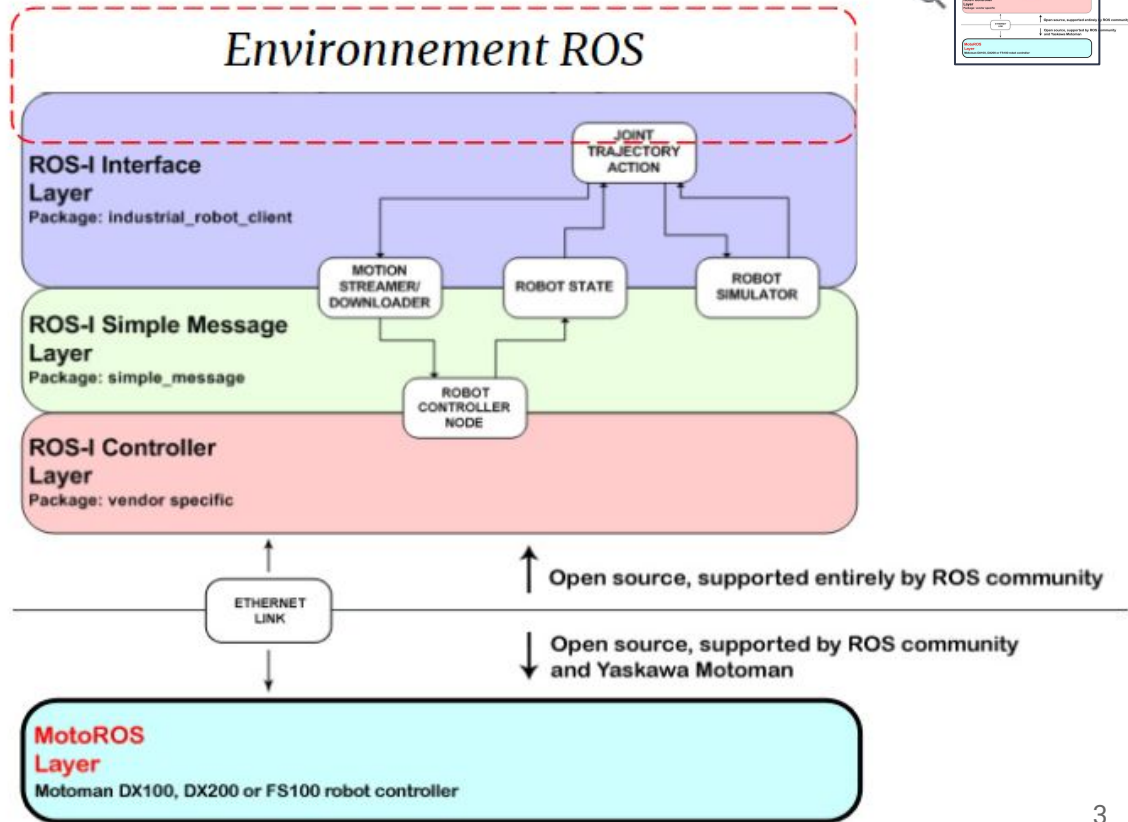
Architecture Simplifiée

Interface normalisée pour  
le contrôle des robots  
industriels

Protocole de Comm. 



Contrôleur YRC1000  
avec MotoROS





Contrôleur YRC1000  
avec MotoROS

## Définition des registres qui vont contenir les valeurs de position et orientation de l'organe terminal

### 6.12 Present Manipulator Position Output Function

#### 6.12.1 Function for Outputting Present Cartesian Position of Manipulator to Register

##### 6.12.1.1 Outline

The present Cartesian position of the manipulator (values in the base coordinates) is output to the specified registers.

##### 6.12.1.2 Parameters

The following parameters specify the details of the function and output register numbers.

S1CxG	Description
208	Enables/Disables the function for outputting the present Cartesian position (in the base coordinates) to registers. (command value) 0: disable 1: enable
209	Specifies the output size to the register. 0: output in 2 bytes 1: output in 4 bytes
210	Cartesian position (command value) X register number of output destination
211	Cartesian position (command value) Y register number of output destination
212	Cartesian position (command value) Z register number of output destination
213	Cartesian position (command value) Rx register number of output destination
214	Cartesian position (command value) Ry register number of output destination
215	Cartesian position (command value) Rz register number of output destination
216	Cartesian position (command value) Re register number of output destination
217	Enables/Disables the function for outputting the present Cartesian position (in the base coordinates) to registers. (FB value) 0: disable 1: enable
218	Specifies the output size to the register. 0: output in 2 bytes 1: output in 4 bytes
219	Cartesian position (FB value) X register number of output destination
220	Cartesian position (FB value) Y register number of output destination
221	Cartesian position (FB value) Z register number of output destination
222	Cartesian position (FB value) Rx register number of output destination
223	Cartesian position (FB value) Ry register number of output destination
224	Cartesian position (FB value) Rz register number of output destination
224	Cartesian position (FB value) Re register number of output destination



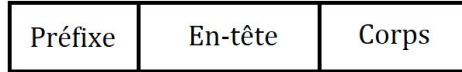
#### <Example 2>

S1C1G	Setting value
217	1
218	1
219	10
220	12
221	14
222	16
223	18
224	20
225	22

When the parameters are set as shown in the above table, the present position is output to the registers as follows:

M010 =	Lower 2 bytes of the	manipulator's present Cartesian position (FB value)	X	[unit: $\mu\text{m}$ ]
M011 =	Upper 2 bytes of the	manipulator's present Cartesian position (FB value)	X	[unit: $\mu\text{m}$ ]
M012 =	Lower 2 bytes of the	manipulator's present Cartesian position (FB value)	Y	[unit: $\mu\text{m}$ ]
M013 =	Upper 2 bytes of the	manipulator's present Cartesian position (FB value)	Y	[unit: $\mu\text{m}$ ]
M014 =	Lower 2 bytes of the	manipulator's present Cartesian position (FB value)	Z	[unit: $\mu\text{m}$ ]
M015 =	Upper 2 bytes of the	manipulator's present Cartesian position (FB value)	Z	[unit: $\mu\text{m}$ ]
M016 =	Lower 2 bytes of the	manipulator's present Cartesian position (FB value)	Rx	[unit: 0.001 deg]
M017 =	Upper 2 bytes of the	manipulator's present Cartesian position (FB value)	Rx	[unit: 0.001 deg]

### Message



- Préfixe = Longueur(En-tête + Corps)

- En-tête =



- Corps = Données

*L'en-tête, divisé en trois champs qui spécifient le type de message envoyé.*

- *msg\_type*: l'identifiant du message.
- *comm\_type*: indique le type de flux des données.
- *reply\_code*: avertit si l'information a été reçue.

*Le corps, ce sont les données.*

ID	Name	Comment
2004	ROS_MSG_MOTO_READ_IO_BIT_REPLY	-
2012	ROS_MSG_MOTO_READ_MREGISTER	Read M-Register (M000 through M999)

Corps: 4 Octets

## Environnement ROS

### Motoman\_memory.h

```
typedef enum
{
    POS_S = 1000010, //M010 = 2 octets inférieurs & M011 = 2 octets supérieurs
    POS_L = 1000012,
    POS_U = 1000014,
    POS_R = 1000016,
    POS_B = 1000018,
    POS_T = 1000020,

    POS_X = 1000030,
    POS_Y = 1000032,
    POS_Z = 1000034,
    POS_Rx = 1000036,
    POS_Ry = 1000038,
    POS_Rz = 1000040,
    POS_Re = 1000042

    POS; // Codé sur 4 octets
}
```

## Mapping de la mémoire

### MREGISTER\_RESERVE\_

Registres en rouge défini par Yaskawa. Ne pas modifier.

Définition	Nom	Description
<b>M010 &amp; M011</b>	MREGISTER_RESERVE_POS_S	Position des axes (FB) [Unité: 0.0001 deg]
<b>M012 &amp; M013</b>	MREGISTER_RESERVE_POS_L	Valeur codée sur 4 octets. Le premier registre correspond aux 2 octets inférieurs. Ex. M010
<b>M014 &amp; M015</b>	MREGISTER_RESERVE_POS_U	
<b>M016 &amp; M017</b>	MREGISTER_RESERVE_POS_R	Le dernier registre correspond aux 2 octets supérieurs. Ex. M011
<b>M018 &amp; M019</b>	MREGISTER_RESERVE_POS_B	
<b>M020 &amp; M021</b>	MREGISTER_RESERVE_POS_T	
<b>M030 &amp; M031</b>	MREGISTER_RESERVE_POS_X	Position cartésienne du manipulateur (FB) X,Y,Z [Unité: µm] Rx,Ry,Rz [Unité: 0.001 deg]
<b>M032 &amp; M033</b>	MREGISTER_RESERVE_POS_Y	
<b>M034 &amp; M035</b>	MREGISTER_RESERVE_POS_Z	
<b>M036 &amp; M037</b>	MREGISTER_RESERVE_POS_Rx	Valeur codée sur 4 octets. Le premier registre correspond aux 2 octets inférieurs. Ex. M030
<b>M038 &amp; M039</b>	MREGISTER_RESERVE_POS_Ry	
<b>M040 &amp; M041</b>	MREGISTER_RESERVE_POS_Rz	Le dernier registre correspond aux 2 octets supérieurs. Ex. M031
<b>M042 &amp; M043</b>	MREGISTER_RESERVE_POS_Re	Valeur du 7eme axe robot (lorsqu'il y en a un). [Unité: 0.001 deg]

## Environnement ROS

```
▼ motoman_msgs
  ▼ msg
    ≡ DynamicJointPoint.msg
    ≡ DynamicJointsGroup.msg
    ≡ DynamicJointState.msg
    ≡ DynamicJointTrajectory.msg
    ≡ DynamicJointTrajectoryFeedback.msg
    ≡ Effort.msg
    ≡ Position.msg
    ≡ Vitesse.msg
  > srv
```

```
motoman > motoman_msgs > msg > ≡ Position.msg
1  #Position des axes en degrés
2  float32 pos_s
3  float32 pos_l
4  float32 pos_u
5  float32 pos_r
6  float32 pos_b
7  float32 pos_t
8  #Position du manipulateur en mm
9  float32 pos_x
10 float32 pos_y
11 float32 pos_z
12 #Orientation du manipulateur en degrés
13 float32 rot_x
14 float32 rot_y
15 float32 rot_z
16 float32 rot_e
17
```