



Plugin user guide

Fieldbus Adapter
v1.0.0

Contents

Copyright notice.....	1
Disclaimers.....	2
Abbreviations.....	3
Plugins.....	4
Getting the plugins for the robot.....	4
Installing plugins.....	5
Updating plugins.....	7
Overview of working with plugins.....	9
Configurations of the Fieldbus Adapter Plugin.....	10
Overview of the Modbus/TCP Specifications.....	11
Features.....	11
Enabling Modbus/TCP.....	11
Units.....	11
Types.....	12
Inputs and outputs.....	12
Input mapping.....	13
Output mapping.....	15
Inputs detailed description.....	17
SessionStatus.....	17
ArmState.....	17
OperatingMode.....	18
ArmCurrent.....	18
ArmVoltage.....	19
AmbientTemp.....	19
SafetyParametersCRC.....	19
SafetyState.....	19
SafetyInputs.....	20
SafetyOutputs.....	20
ProgramRunnerStatus.....	21
ProgramRunnerInputBits.....	21
ProgramExecutionStatus.....	22
ProgramID.....	22
JointPositions.....	23
JointVelocities.....	23
JointTorques.....	24
JointMotorCurrents.....	24

JointMotorTemperatures.....	24
TcpPositionX.....	25
TcpPositionY.....	25
TcpPositionZ.....	25
TcpOrientationX.....	26
TcpOrientationY.....	26
TcpOrientationZ.....	26
TcpPositionOffsetX.....	26
TcpPositionOffsetY.....	27
TcpPositionOffsetZ.....	27
TcpOrientationOffsetX.....	27
TcpOrientationOffsetY.....	27
TcpOrientationOffsetZ.....	28
TcpTranslationSpeedX.....	28
TcpTranslationSpeedY.....	28
TcpTranslationSpeedZ.....	28
TcpRotationSpeedX.....	29
TcpRotationSpeedY.....	29
TcpRotationSpeedZ.....	29
TcpForceX.....	29
TcpForceY.....	30
TcpForceZ.....	30
TcpTorqueX.....	30
TcpTorqueY.....	30
TcpTorqueZ.....	31
GravityVectorX.....	31
GravityVectorY.....	31
GravityVectorZ.....	31
Payload.....	32
PayloadCoMX.....	32
PayloadCoMY.....	32
PayloadCoMZ.....	32
DigitalInputs.....	33
DigitalFeedbackM8.....	33
AnalogInput1.....	33
AnalogInput2.....	34
AnalogInput3.....	34
AnalogInput4.....	35
AnalogInput1M8A.....	35
AnalogInput2M8A.....	36
AnalogInput1M8B.....	36
AnalogInput2M8B.....	37
ModelID.....	37
NumberOfJoints.....	38
InputUserVariables.....	38

Outputs detailed description.....	39
DigitalOutput.....	39
DigitalOutputWriteMask.....	39
DigitalOutputM8.....	39

DigitalOutputM8WriteMask.....	41
AnalogOutput1.....	41
AnalogOutput2.....	42
AnalogOutput3.....	42
AnalogOutput4.....	43
AnalogOutput1M8A.....	43
AnalogOutput2M8A.....	44
AnalogOutput1M8B.....	44
AnalogOutput2M8B.....	44
RobotControl.....	45
DesiredProgramID.....	46
ProgramRunnerBits.....	47
OutputUserVariables.....	48

Glossary.....	49
B.....	49
C.....	49
D.....	49
E.....	49
F.....	49
G.....	49
H.....	50
I.....	50
K.....	50
L.....	50
M.....	50
O.....	51
P.....	51
R.....	51
S.....	51
T.....	52
U.....	52

Copyright notice

The information in the Fieldbus Adapter Plugin user guide contains proprietary information and, as such, no part may be reproduced, in part or in whole, or transmitted in any form or by any means, electronically or mechanically, including photocopying, recording, or any information or storage retrieval system, without prior permission in writing from Kinova.

Disclaimers

Kinova[®] and the Kinova logo are registered trademarks of Kinova inc., herein referred to as Kinova.

Kortex[™] is a trademark of Kinova inc.

All other brand and product names are trademarks or registered trademarks of their respective owners.

The Fieldbus Adapter Plugin is an evolving product. Kinova continually works towards improving the reliability and performance of its products. Kinova does its best to ensure the content of the Fieldbus Adapter Plugin user guide is accurate and precise with respect to the latest version of the robot. However, Kinova does not take responsibility for any erroneous, inaccurate, or missing information.



Note: The user guide may be updated from time to time. Please make sure to look at the latest version on our website.

The version of this document is in the header of the pages of the document.

Any questions or comments concerning this document, the information it contains or the product it describes may be addressed through support@kinova.ca.

Abbreviations

BE	big-endian
DoF	degrees of freedom
GUI	Graphical User Interface
HMI	human machine interface
LE	little-endian
MCU	Main Control Unit
MPU	Main Processing Unit
PLC	programmable logic controller
SCU	Safety Control Unit
TCP	Tool Center Point

Plugins

The modular design of Link 6 makes it a robot with expanding capabilities. The expanding capabilities are added on as [plugins](#).

All plugins start to run right after they are installed. However, they can be stopped so the settings can be modified. All plugins that are running are, by default, active. They can be deactivated to stop their functionality.

The Fieldbus Adapter Plugin does not contain actions that the user can use when programming the robot with the visual programming interface in Kortex Web App. That is because the Fieldbus Adapter Plugin is a communication plugin.



Important: One and only one Fieldbus Adapter Plugin can be installed and running on the robot at a time.

All plugins have two pages: **Configurations** and **About**.

Getting the plugins for the robot

Upload and installation must be done from a computer and a [USB](#) storage device.

Before you begin

All plugins have dependencies on different aspects of the robot.

Make sure to verify all dependencies before installing the plugin.

Table 1: Plugin dependencies and expected issues when dependencies not met

Dependency	What to expect when dependencies are not met
Robot model must be Link 6	Installation failure
Software package must be compatible with the robot	Installation failure

About this task

The plugin installation file is named `fieldbus_adapter_<version>.kp`, where `<version>` is the version of the plugin. Contact Kinova to get the appropriate plugin file if you are unsure which files you need.

A computer or USB device is needed to upload and install the plugin installation file.



Note: Plugins are updated from time to time. Always make sure you have the latest plugin for your installation.

Procedure

1. Download the plugin installation file `fieldbus_adapter_<version>.kp`.
2. Save the plugin installation file in the root directory of the USB storage device.
3. Eject the USB after the transfer of the plugin installation file is complete.
4. Plug the USB storage device on the [controller](#).
5. Switch on the [teach pendant](#).
6. Enter the administrator credentials.

7. Tap **CONNECT**.

8. Tap **POWER ON**.

The arm is switched on.

What to do next

Install the required plugins from the **Plugins** page in [Kortex Web App](#).

Installing plugins

The Fieldbus Adapter Plugin must be installed when there are improvements to the communication interface.

Before you begin

The plugin files must be on a USB storage device. The USB must be plugged into the controller

About this task

Access all installed plugins by tapping **Systems > Plugins**.

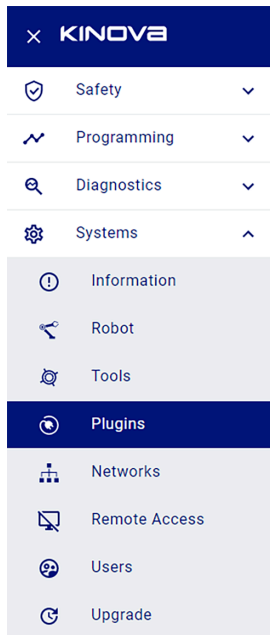
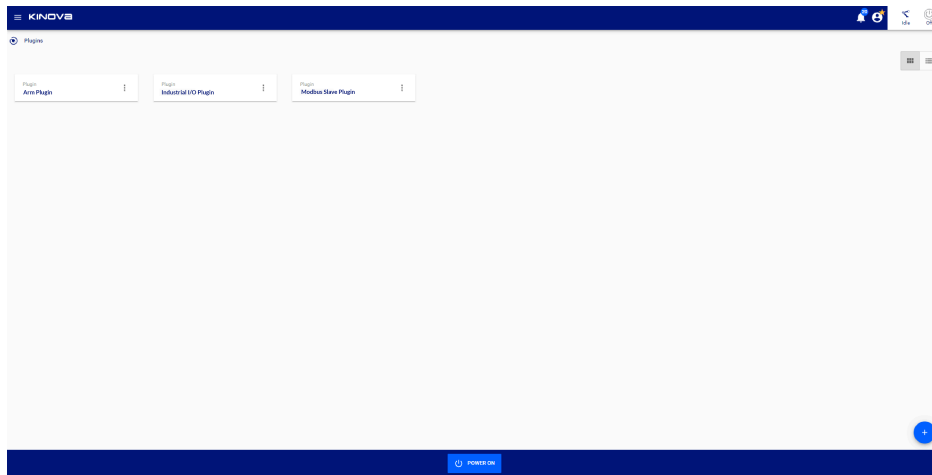


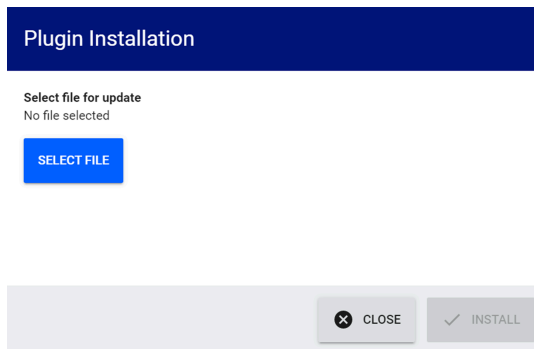
Figure 1: Accessing the plugins

Procedure

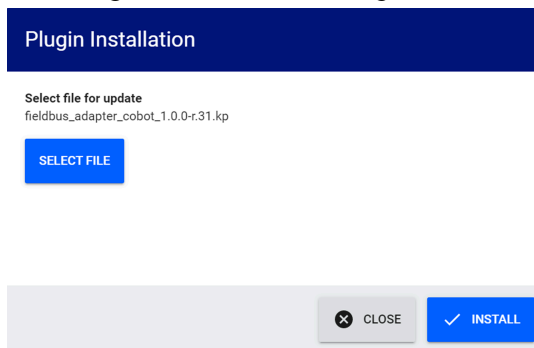
1. Tap **Systems > Plugins**.
The **Plugins** page launches.



2. Tap **+**.
A **Plugin Installation** dialog launches.

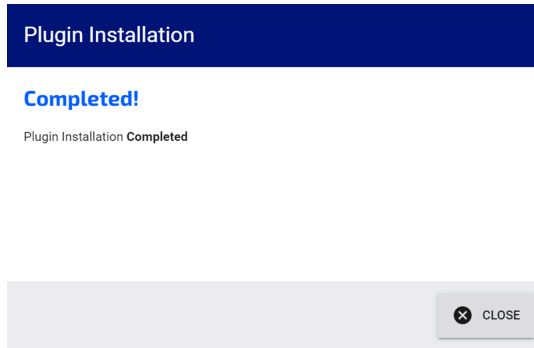


3. Tap **SELECT FILE**.
A file browse window launches.
4. Browse to the plugin installation files.
5. Select a plugin installation file.
6. Tap **Open**.
The **Plugin Installation** dialog shows the name of the selected file.



7. Tap **INSTALL**.
The **Plugin Installation** dialog shows that the plugin installation is in progress.

8. Tap **CLOSE** when installation is complete.



Results

The installed plugin appears on the **Plugins** page. A running icon (Plugin ) displayed near the plugin name on the **Plugins** page, indicates the plugin is running.

Updating plugins

Integrate improvements to the plugins by updating them.

Before you begin

Download the latest version of the plugin from the Contact Kinova to get the appropriate plugin file if you are unsure which files you need.

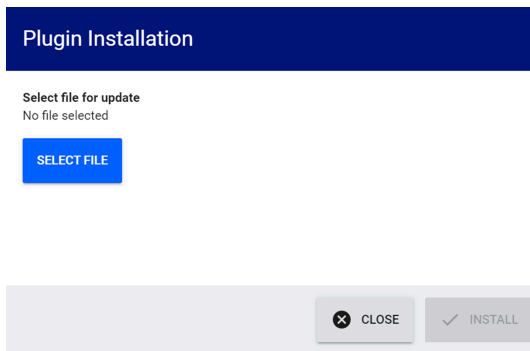
About this task

Updating plugins is similar to installing them. Use a computer or USB device.

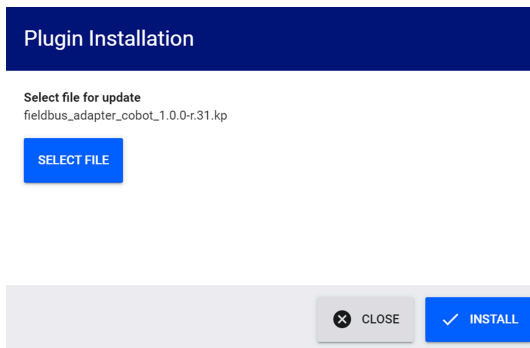
Procedure

1. Get the latest version of the plugin.
2. Tap **Systems > Plugins**.
3. Tap the plugin that you want to update.
4. Tap **Stop** (Stop )
5. Tap **Deactivate**.
6. Navigate back to the **Plugins** page.
7. Tap the kebab menu () of the plugin that you want to update.
8. Tap **Delete**.

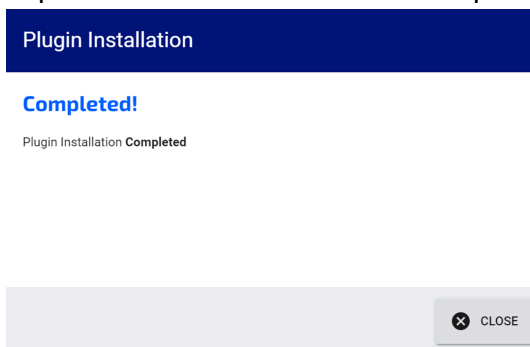
9. Tap **+**.
A **Plugin Installation** dialog launches.



10. Tap **SELECT FILE**.
A file browse window launches.
11. Browse to the plugin installation files.
12. Select a plugin installation file.
13. Tap **Open**.
The **Plugin Installation** dialog shows the name of the selected file.



14. Tap **INSTALL**.
The **Plugin Installation** dialog shows that the plugin installation is in progress.
15. Tap **CLOSE** when installation is complete.



Results

The installed plugin appears on the **Plugins** page. A running icon (Plugin ) displayed near the plugin name on the **Plugins** page, indicates the plugin is running.

Overview of working with plugins

Tapping a plugin from the **Plugins** page launches the page of that plugin.

Figure 2: Example of a plugin page of a plugin that is stopped

- 1 Tap **Plugins** to return to the list of plugins.
- 2 Name of the selected plugin with the name of the namespace associated with the selected plugin in parentheses.
- 3 Tap **Configurations** to view and modify the current configurations of the selected plugin.



Important: Tap **Stop** (Stop ■) before you attempt to modify the current configurations.



Important: Tap **Run** (Run ►) after you modify the configurations. The plugin runs with the latest configuration values.

When there are no modifiable configurations in the plugin, a message is displayed.

This plugin does not expose any configuration.

- 4 Tap **About** to view important information about the selected plugin.
 - Name
 - Package Name
 - Version
 - Installation Type
 - Description
- 5 Tap **Deactivate** to deactivate plugins that need to be installed but are not being used.
Tap **Activate Plugin** to activate a deactivated plugin.
- 6 The current operating state of the plugin.
Tap **Stop** (Stop ■) to stop the plugin from running. The button changes to **Run** (Run ►).
Tap **Run** to restart the plugin after modifying its settings. The button changes to **Stop**.

Configurations of the Fieldbus Adapter Plugin

The **Configurations** page of the Fieldbus Adapter Plugin displays the variables that are used in the communication interface.

To access the **Configurations** of the Fieldbus Adapter Plugin plugin, tap **Systems > Plugins > Fieldbus Adapter Plugin**.



Important: Tap **Stop** (Stop icon) before you attempt to modify the current configurations.



Important: Tap **Run** (Run icon) after you modify the configurations. The plugin runs with the latest configuration values.



Remember: In the case of the Fieldbus Adapter Plugin, configurations are variables. Each of the variables can take on one of the defined types.

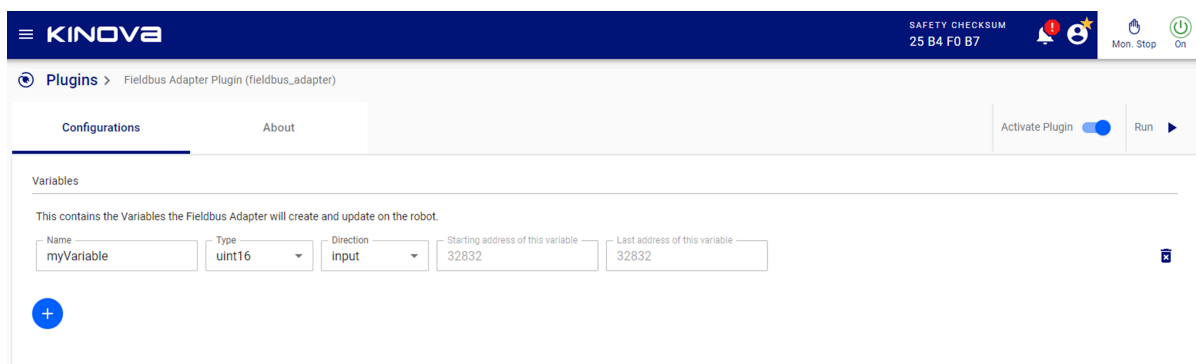


Figure 3: Configurations page of the Fieldbus Adapter Plugin

Table 2: Working with the Configuration page of the Fieldbus Adapter Plugin

Item	Description
	Tap to add a new variable that can be used.
Name	Enter a meaningful name for the variable.
Type	Select a type from the drop-down menu.
Direction	Select either Input or Output .
Starting address of this variable	The input variable is assigned an address between 0X0066 and 0x1065. The output variable is assigned an address between 0x8040 and 0x903F.
Last address of this variable	The variable takes a specific amount of space in the registers. The last address is determined by taking the starting address and adding to it the size of its type.
	Tap to delete the variable.

Overview of the Modbus/TCP Specifications

Link 6 has a Modbus/TCP communication interface that can be used to monitor the status of the robot. The interface can also be used for other industrial use cases where Link 6 is controlled by a [PLC](#) or other industrial supervisory system.

Features

The Modbus/TCP communication interface is used in multiple applications where communication within the robot needs to be customized.

Modbus Standard	V1.1b3
Port	502
Number of connections	Single connection
Modbus function codes	• 3: Read holding register • 6: Write single holding register • 16: Write multiple holding registers • 23: Read-write multiple holding registers
Applications	• Monitoring • Program execution control • I/O control • Application-specific control through user variables

Enabling Modbus/TCP

There are a few things to check for in Kortex Web App to make sure that Modbus/TCP is enabled and running on Link 6.

- Tap **Systems > Plugins**.
 - **Fieldbus Adapter Plugin** is available.
- Tap **Fieldbus Adapter Plugin**.
 - The **Configurations** page of the plugin is grayed out.
 - The **Stop** button is enabled.
- Tap **Systems > Remote Access**.
 - The **Modbus Server** option is on.

Units

The Modbus interface uses measurements of degrees, current, and velocity among others.

Abbreviation	Description
mA	milliamps
mV	millivolts
°C/10	tenths of degrees Celsius

Abbreviation	Description	
°/10	tenths of degrees	
°/100	hundredths of degrees	
°/10/s	tenths of degrees per second	
mm	millimeters	
mm/s	millimeters per second	
mm/s ²	millimeters per second squared	
mm/10	tenths of mm	
g	grams	
N·m/100	hundredths of Newton-meters	
N/100	hundredths of Newtons	
analog	Pin is configured as voltage input	V/6553
	Pin is configured as current input	mA/2621
	Pin is configured as voltage output	V/744
	Pin is configured as current output	mA/327

Types

Each elementary type used by the Modbus/TCP interface uses different widths in the register.

Type	Register width	Description
UINT16	1	Unsigned 16 bits integer
INT16	1	Signed 16 bits integer
UINT32	2	Unsigned 32 bits integer
INT32	2	Signed 32 bits integer
FLOAT	2	IEEE 754 32 bits floating point
JOINT_DATA	7	An array of 7 INT16

For types spanning across two registers, data is encoded using the *Modicon compatibility mode*; the low address register contains the smaller bits and the high address register contains the larger bits. The registers themselves, however, are *big-endian*. The 4 bytes of a 32-bits type are organized in a specific manner.

Modbus
Address

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
n	byte 0								byte 1							
n + 1	byte 2								byte 3							

Inputs and outputs

The Fieldbus Adapter Plugin uses zero-based indexes for addresses and bit addressing.



Important: *Input* refers to an input of the PLC or to an output of the robot. *Output* refers to an output of the PLC or to an input of the robot.

Although it is possible to write to the input registers, writing to them has no effect on the robot and the value read back is the original value from the robot.

Input mapping

Each input consists of 16 bits and uses one or more bits in the Modbus HOLDING REGISTER address space. Inputs come from the robot and go to the PLC.

Modbus address	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0000	SessionStatus															
0x0001	ArmState															
0x0002	OperatingMode															
0x0003	ArmCurrent															
0x0004	ArmVoltage															
0x0005	AmbientTemp															
0x0006	SafetyParametersCRC															
0x0007																
0x0008	SafetyState															
0x0009	SafetyInputs															
0x000A	SafetyOutputs															
0x000B	ProgrammerRunnerStatus															
0x000C	ProgramRunnerInputBits															
0x000D	ProgrammerExecutionStatus															
0x000E	ProgramID															
0x000F																
0x0010	JointPositions															
...																
0x0016																
0x0017	JointVelocities															
...																
0x001D																
0x001E	JointTorques															
...																
0x0024																
0x0025	JointMotorCurrents															
...																
0x002B																

0x002C	
...	JointMotorTemperatures
0x0032	
0x0033	TcpPositionX
0x0034	TcpPositionY
0x0035	TcpPositionZ
0x0036	TcpOrientationX
0x0037	TcpOrientationY
0x0038	TcpOrientationZ
0x0039	TcpPositionOffsetX
0x003A	TcpPositionOffsetY
0x003B	TcpPositionOffsetZ
0x003C	TcpOrientationOffsetX
0x003D	TcpOrientationOffsetY
0x003E	TcpOrientationOffsetZ
0x003F	TcpTranslationSpeedX
0x0040	TcpTranslationSpeedY
0x0041	TcpTranslationSpeedZ
0x0042	TcpRotationSpeedX
0x0043	TcpRotationSpeedY
0x0044	TcpRotationSpeedZ
0x0045	TcpForceX
0x0046	TcpForceY
0x0047	TcpForceZ
0x0048	TcpTorqueX
0x0049	TcpTorqueY
0x004A	TcpTorqueZ
0x004B	GravityVectorX
0x004C	GravityVectorY
0x004D	GravityVectorX
0x004E	Payload
0x004F	PayloadCoMX
0x0050	PayloadCoMY
0x0051	PayloadCoMZ
0x0052	DigitalInputs

0x0053	DigitalFeedbackM8
0x0054	AnalogInput1
0x0055	AnalogInput2
0x0056	AnalogInput3
0x0057	AnalogInput4
0x0058	AnalogInput1M8A
0x0059	AnalogInput2M8A
0x005A	AnalogInput1M8B
0x005B	AnalogInput2M8B
0x005C	
...	
0x0063	
0x0064	ModelID
0x0065	NumberOfJoints
0x0066	
...	InputUserVariables
0x1065	

Output mapping

Each output consists of 16 bits and uses one or more bits in the Modbus **HOLDING REGISTER** address space. Outputs come from the PLC and go to the robot.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x8000	DigitalOutput															
0x8001	DigitalOutputWriteMask															
0x8002	DigitalOutputM8															
0x8003	DigitalOutputM8WriteMask															
0x8004	AnalogOutput1															
0x8005	AnalogOutput2															
0x8006	AnalogOutput3															
0x8007	AnalogOutput4															
0x8008	AnalogOutput1M8A															
0x8009	AnalogOutput2M8A															
0x0800A	AnalogOutput1M8B															
0x0800B	AnalogOutput2M8B															
0x0800C																

...	
0x0801F	
0x8020	RobotControl
0x8021	
0x8022	DesiredProgramID
0x8023	
0x8024	ProgramRunnerBits
0x8025	
...	
0x803F	
0x8040	
...	OutputUserVariables
0x903F	

Inputs detailed description

Each input register has a unique function and location in the `HOLDING_REGISTERS` address space.

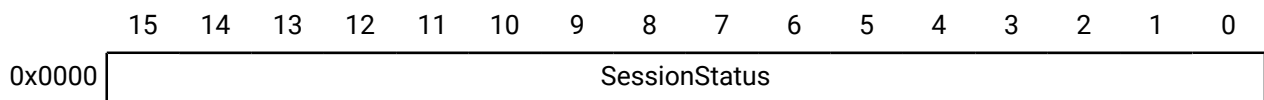
SessionStatus

When single source control is active and a part of the robot that has higher priority, such as when Kortex Web App or the teach pendant starts up, the value in the register is 0; the client has no control over the robot. At all other times, even when single source control is not active, the value is 1.

SessionStatus register type UINT16

Address 0x0000

Unit bitfield



ArmState

The `ArmState` register represents the state of the arm. It is possible to control the state of the arm through the `RobotControl` register. The arm can move when this register has the value `ARMSTATE_ARM_OPERATIONAL` or `ARMSTATE_RECOVERY`.

ArmState register type UINT16

Address 0x0001

Unit enumeration

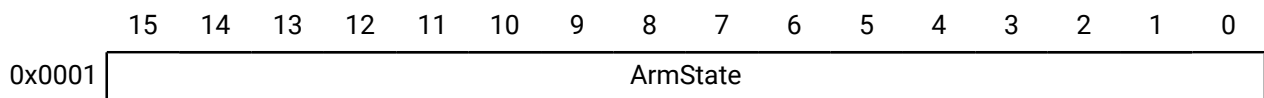


Table 3: Enumeration of the ArmState register

Value	Description
0	ARMSTATE_UNSPECIFIED
1	ARMSTATE_BASE_INITIALIZATION
2	ARMSTATE_IDLE
3	ARMSTATE_INITIALIZATION
4	ARMSTATE_IN_FAULT
5	ARMSTATE_MAINTENANCE
6	ARMSTATE_SERVOING_LOW_LEVEL
7	ARMSTATE_SERVOING_READY
8	ARMSTATE_SERVOING_PLAYING_SEQUENCE

Value	Description
9	ARMSTATE_SERVOING_MANUALLY_CONTROLLED
10	ARMSTATE_BRAKE_ON
11	ARMSTATE_POSITION_UNKNOWN
12	ARMSTATE_ARM_OPERATIONAL
13	ARMSTATE_RECOVERY
14	ARMSTATE_CONTROLLER_INITIALIZATION
15	ARMSTATE_WAIT_FOR_POSITION_CONFIRMATION
16	ARMSTATE_IN_FAULT_POWERED_OFF
17	ARMSTATE_IN_UNRECOVERABLE_FAULT

OperatingMode

The value in the register reflects the current way in which the user is interacting with the robot. The current operating mode is reflected in the lights on the wrist ring as well as in the Robot control panel in Kortex Web App.

OperatingMode register type UINT16
 Address 0x0002
 Unit enumeration

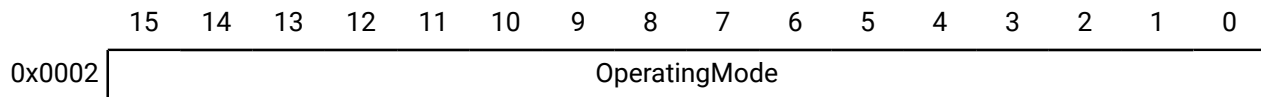


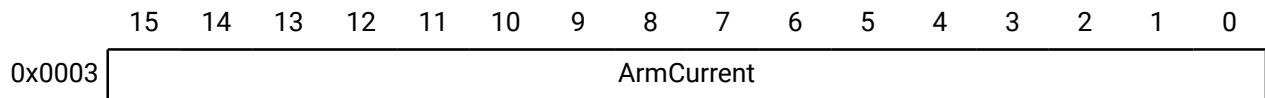
Table 4: Enumeration of the OperatingMode register

Value	Description
0	OPERATING_MODE_UNSPECIFIED
1	OPERATING_MODE_JOG_MANUAL
2	OPERATING_MODE_HAND_GUIDING
3	OPERATING_MODE_HOLD_TO_RUN
4	OPERATING_MODE_AUTO
5	OPERATING_MODE_MONITORED_STOP

ArmCurrent

The value in the register is the amount of current going through the arm at that moment.

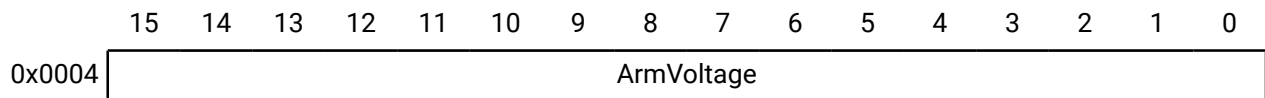
ArmCurrent register type UINT16
 Address 0x0003
 Unit mA



ArmVoltage

The value in the register is the amount of voltage in the arm.

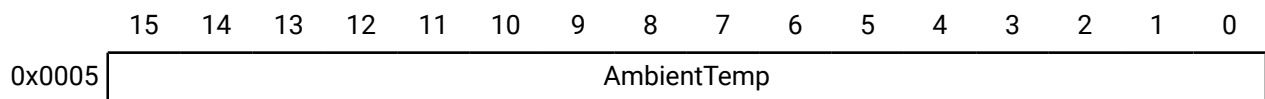
ArmVoltage register type	UINT16
Address	0x0004
Unit	mV



AmbientTemp

The value in the register reflects the current temperature of the controller.

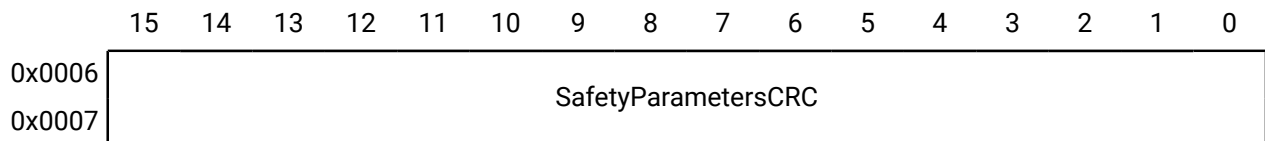
AmbientTemp register type	UINT16
Address	0x0005
Unit	°C/10



SafetyParametersCRC

The register holds a unique number that represents the current state of several different safety elements that are derived from the [MCU](#) and the [SCU](#).

SafetyParametersCRC register type	UINT32
Address	0x0006
Unit	integer



SafetyState

Each bit in the register, set with a value of 0 or 1, reflects which safety functions are triggered.

SafetyState register type	UINT16
Address	0x0008

Unit	bitfield															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0008							TEFMT	TOMT	TESMT	PZMT	MST	JSMT	JPMT	PST	EST	UM

Table 5: The bits of the SafetyState register

Bit	Description
TEFMT	TCP-Elbow Force Monitoring triggered
TOMT	TCP Orientation Monitoring triggered
TESMT	TCP-Elbow Speed Monitoring triggered
PZMT	Protection Zone Monitoring triggered
MST	Monitored Stop triggered
JSMT	Joint Speed Monitoring triggered
JPMT	Joint Position Monitoring triggered
PST	Protective Stop triggered
EST	Emergency Stop triggered
UM	Under Monitoring

SafetyInputs

The eight least significant bits of the register, from 0 to 7, represent the state of the safety function associated with each input channel on the **Safety I/Os** page in Kortex Web App .



Remember: Access the safety inputs on the **Safety I/Os** page by tapping **Safety > Safety I/Os > Inputs**.

SafetyInputs register type	UINT16
Address	0x0009
Unit	bitfield

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0009	SafetyInputs															

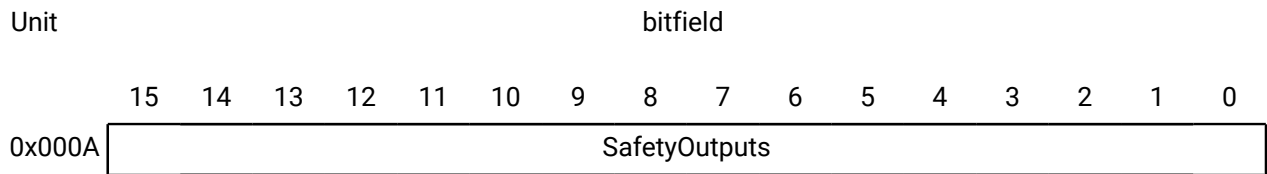
SafetyOutputs

The eight least significant bits of the register, from 0 to 7, represent the state of the safety function associated with each output channel on the **Safety I/Os** page in Kortex Web App .



Remember: Access the safety outputs on the **Safety I/Os** page by tapping **Safety > Safety I/Os > Outputs**.

SafetyInputs register type	UINT16
Address	0x000A



ProgramRunnerStatus

The register holds the current status of the program.



Remember: Programs are selected from the **Programs** page in Kortex Web App.

ProgramRunnerStatus register type UINT16
 Address 0x000B
 Unit enumeration

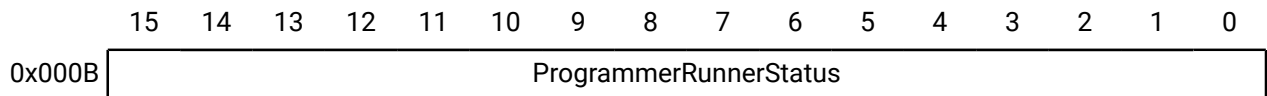


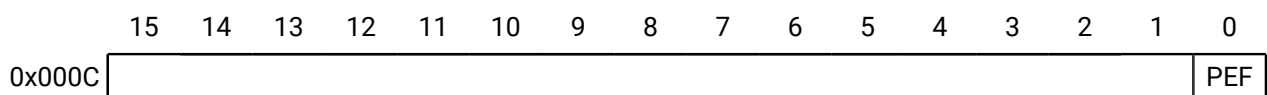
Table 6: Enumeration of the ProgramRunnerStatus register

Value	Description
0	STATUS_UNSPECIFIED
1	STATUS_IDLE
2	STATUS_RUNNING
3	STATUS_PAUSED
4	STATUS_STARTING
5	STATUS_STOPPING
6	STATUS_PAUSED_AUTOMATIC_RESUME
7	STATUS_WAITING_FOR_ACKNOWLEDGE

ProgramRunnerInputBits

The flag in the least significant bit of the register represents whether a program is running. 0 indicates a program is running. 1 indicates the program is in one of three states after running a program: stopped, completed, failed.

ProgramRunnerInputBits register type UINT16
 Address 0x000C
 Unit bitfield



The `ProgramRunnerInputBits` register is a bitfield.

Table 7: The bits of the `ProgramRunnerInputBits` register

Bit	Description
PEF	- The bit is raised when a program finishes. - The bit is lowered when a program starts. - The bit is lowered when the <code>ACK_PEF</code> bit is set. - Clear this bit with the <code>ACK_PEF</code> bit at the start of every robot program for more robust control while running the program. - Use this bit to confirm that a robot program is finished running.

ProgramExecutionStatus

The register holds the current status of the program that is selected to run.

<code>ProgramExecutionStatus</code> register type	UINT16
Address	0x000D
Unit	enumeration

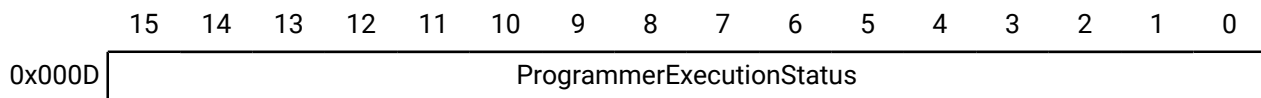


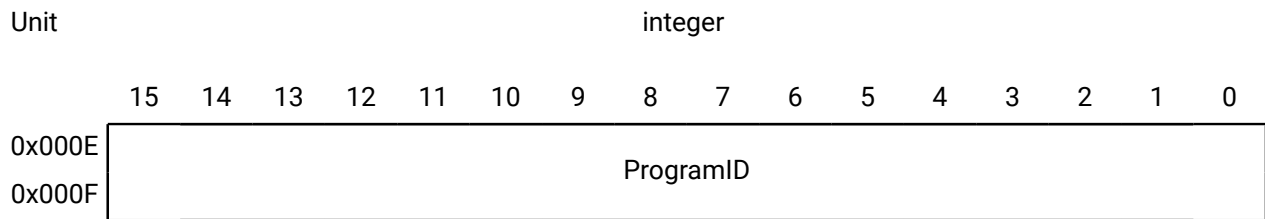
Table 8: Enumeration of the `ProgramExecutionStatus` register

Value	Description
0	EXECUTION_EVENT_UNSPECIFIED
1	EXECUTION_EVENT_COMPLETED
2	EXECUTION_EVENT_FAILED
3	EXECUTION_EVENT_FEEDBACK
4	EXECUTION_EVENT_PAUSED
5	EXECUTION_EVENT_RESUMED
6	EXECUTION_EVENT_STARTED
7	EXECUTION_EVENT_STOPPED

ProgramID

Each program that is created in Kortex Web App is associated with a number. The number is stored in the `ProgramID` register. When you tap on a program in the **Programs** page or when you create a program, the number is displayed in parentheses as part of the name of the program.

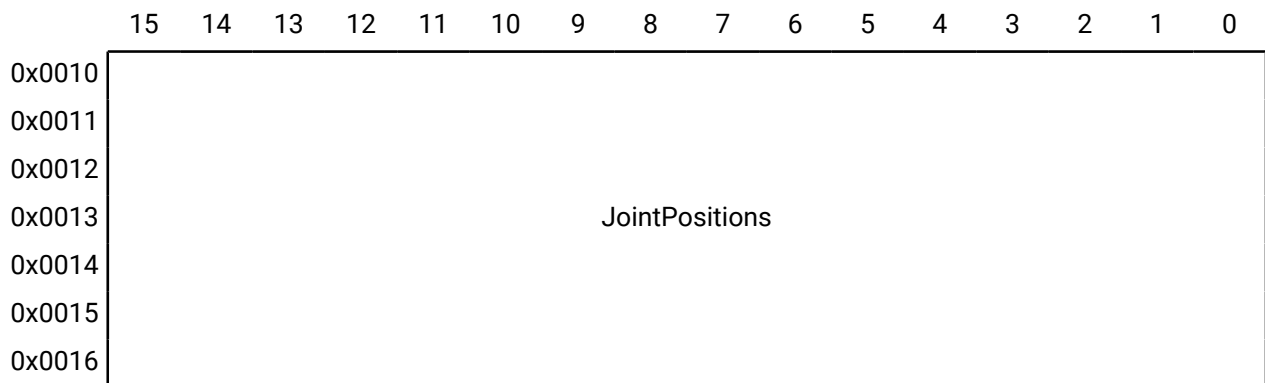
<code>ProgramID</code> register type	UINT32
Address	0x000E



JointPositions

Several consecutive registers represent the current position of each joint from the first to the last.

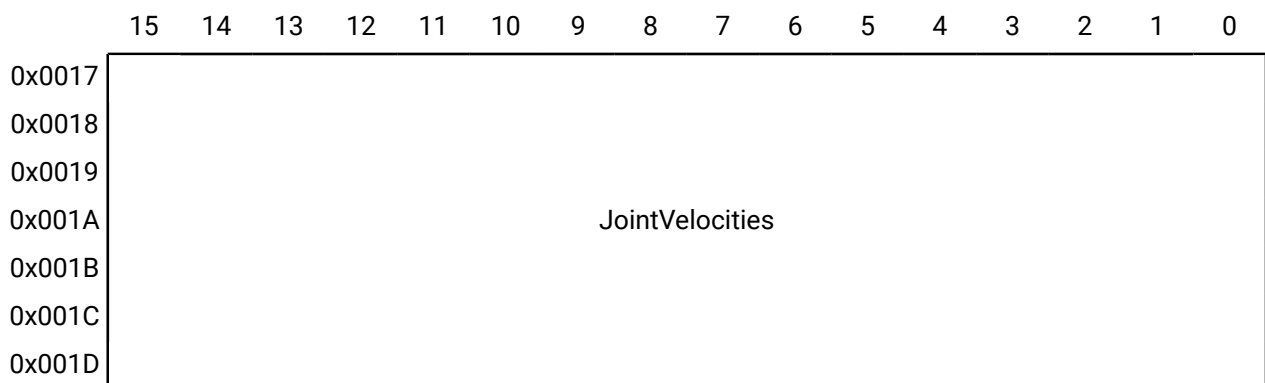
JointPositions register type JOINT_DATA
 Address 0x0010
 Unit °/10



JointVelocities

Several consecutive registers represent the current velocity of each joint from the first to the last.

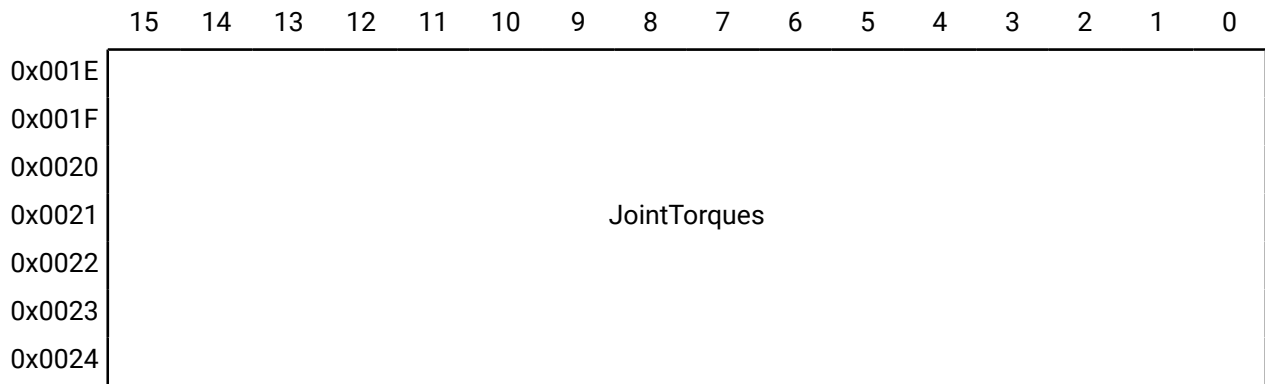
JointPositions register type JOINT_DATA
 Address 0x0017
 Unit °/10/s



JointTorques

Several consecutive registers represent the current torque of each joint from the first to the last.

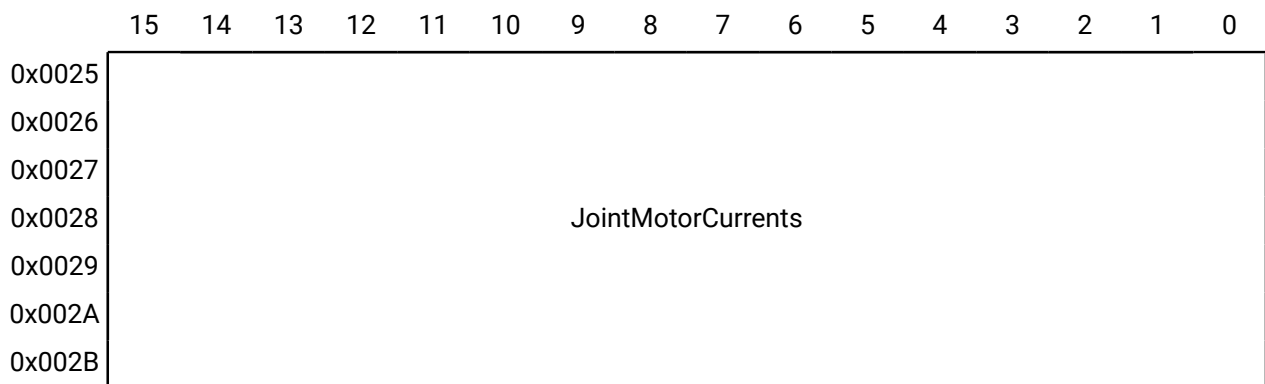
JointTorques register type	JOINT_DATA
Address	0x001E
Unit	N·m/100



JointMotorCurrents

Several consecutive registers represent the current of each motor in each joint from the first to the last.

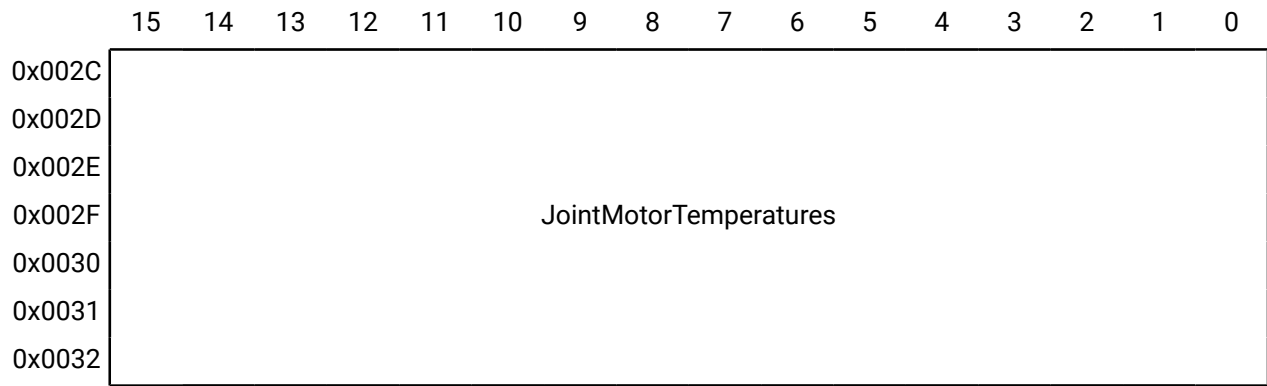
JointMotorCurrents register type	JOINT_DATA
Address	0x0025
Unit	mA



JointMotorTemperatures

Several consecutive registers represent the current temperature of each motor in each joint from the first to the last.

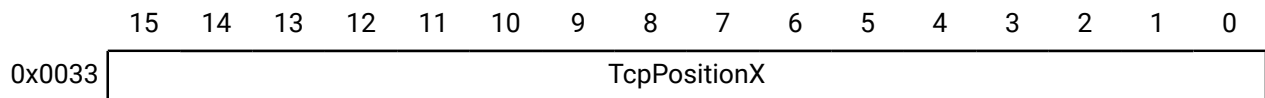
JointMotorTemperatures register type	JOINT_DATA
Address	0x002C
Unit	°C/10



TcpPositionX

The register holds the X component of the position of the TCP.

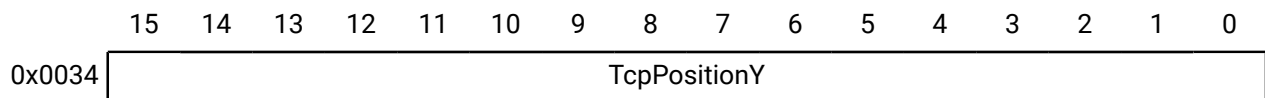
TcpPositionX register type INT16
 Address 0x0033
 Unit mm/10



TcpPositionY

The register holds the Y component of the position of the TCP.

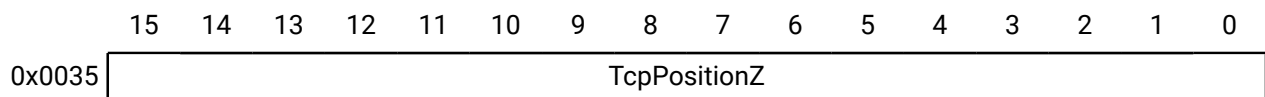
TcpPositionY register type INT16
 Address 0x0034
 Unit mm/10



TcpPositionZ

The register holds the Z component of the position of the TCP.

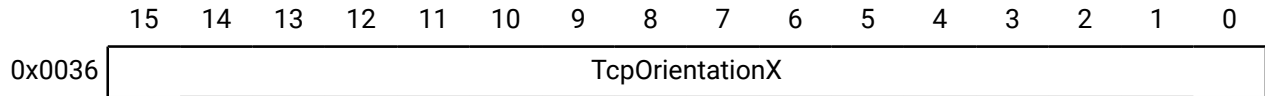
TcpPositionZ register type INT16
 Address 0x0035
 Unit mm/10



TcpOrientationX

The register holds the X component of the orientation of the TCP.

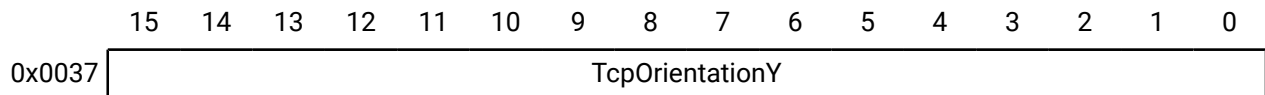
TcpOrientationX register type INT16
 Address 0x0036
 Unit °/100



TcpOrientationY

The register holds the Y component of the orientation of the TCP.

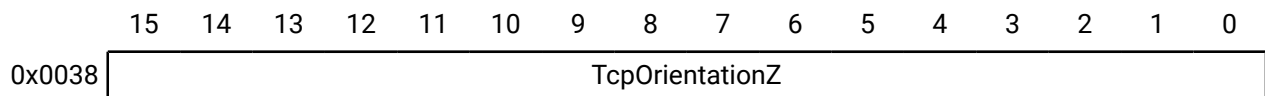
TcpOrientationY register type INT16
 Address 0x0037
 Unit °/100



TcpOrientationZ

The register holds the Z component of the orientation of the TCP.

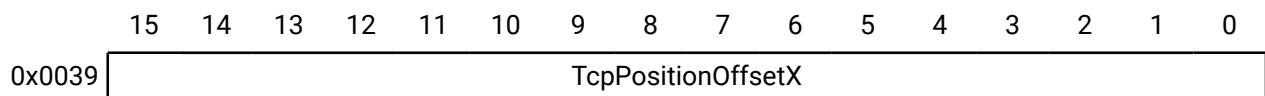
TcpOrientationZ register type INT16
 Address 0x0038
 Unit °/100



TcpPositionOffsetX

The register holds the X component of the position offset of the TCP.

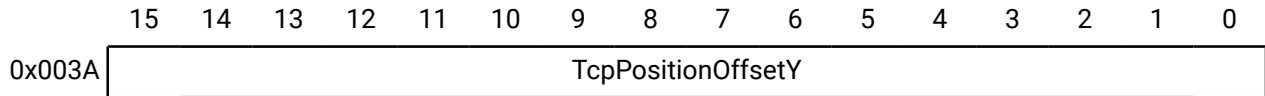
TcpPositionOffsetX register type INT16
 Address 0x0039
 Unit mm/10



TcpPositionOffsetY

The register holds the Y component of the position offset of the TCP.

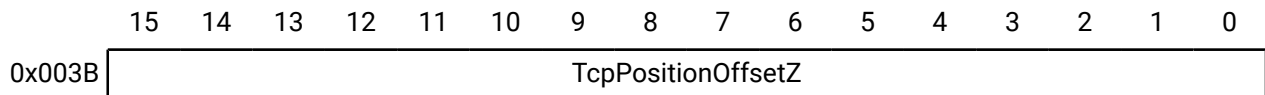
TcpPositionOffsetY register type	INT16
Address	0x003A
Unit	mm/10



TcpPositionOffsetZ

The register holds the Z component of the position offset of the TCP.

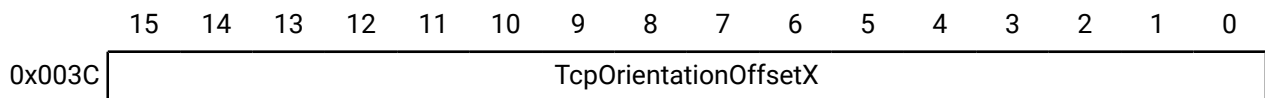
TcpPositionOffsetZ register type	INT16
Address	0x003B
Unit	mm/10



TcpOrientationOffsetX

The register holds the X component of the orientation offset of the TCP.

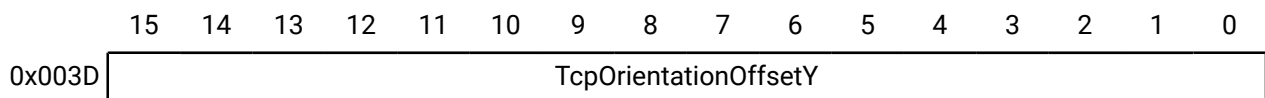
TcpOrientationOffsetX register type	INT16
Address	0x003C
Unit	°/100



TcpOrientationOffsetY

The register holds the Y component of the orientation offset of the TCP.

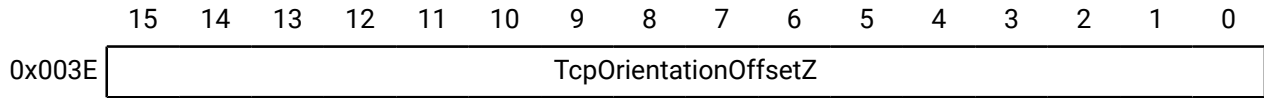
TcpOrientationOffsetY register type	INT16
Address	0x003D
Unit	°/100



TcpOrientationOffsetZ

The register holds the Z component of the orientation offset of the TCP.

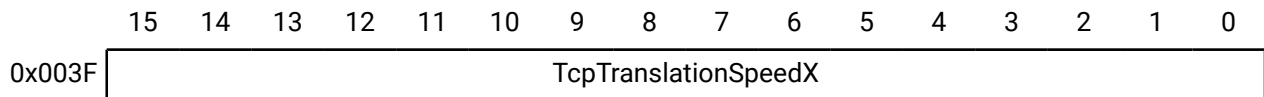
TcpOrientationOffsetZ register type	INT16
Address	0x003E
Unit	°/100



TcpTranslationSpeedX

The register holds the X component of the translation speed of the TCP.

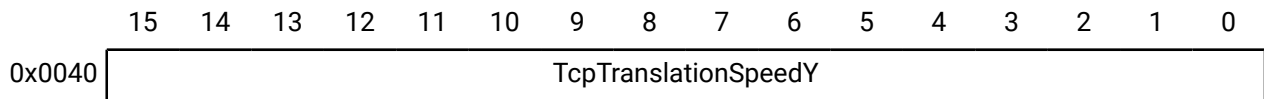
TcpTranslationSpeedX register type	INT16
Address	0x003F
Unit	mm/s



TcpTranslationSpeedY

The register holds the Y component of the translation speed of the TCP.

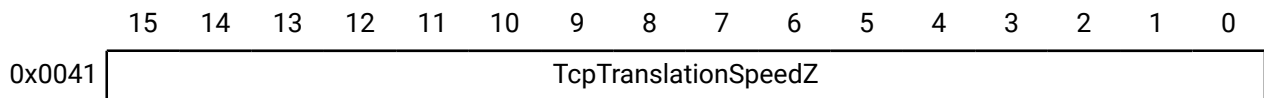
TcpTranslationSpeedY register type	INT16
Address	0x0040
Unit	mm/s



TcpTranslationSpeedZ

The register holds the Z component of the translation speed of the TCP.

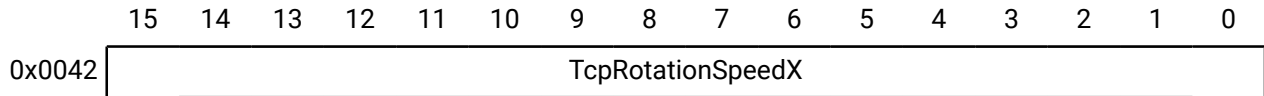
TcpTranslationSpeedZ register type	INT16
Address	0x0041
Unit	mm/s



TcpRotationSpeedX

The register holds the X component of the rotational speed of the TCP.

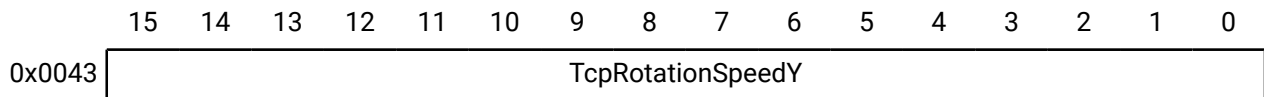
TcpTranslationSpeedZ register type INT16
 Address 0x0042
 Unit °/10/s



TcpRotationSpeedY

The register holds the Y component of the rotational speed of the TCP.

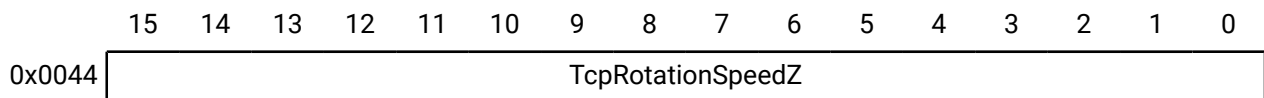
TcpRotationSpeedY register type INT16
 Address 0x0043
 Unit °/10/s



TcpRotationSpeedZ

The register holds the Z component of the rotational speed of the TCP.

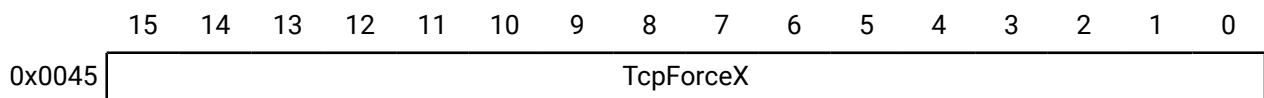
TcpRotationSpeedZ register type INT16
 Address 0x0044
 Unit °/10/s



TcpForceX

The register holds the X component of the force used by the TCP.

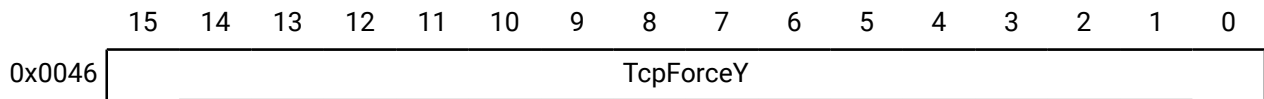
TcpForceX register type INT16
 Address 0x0045
 Unit N/100



TcpForceY

The register holds the Y component of the force used by the TCP.

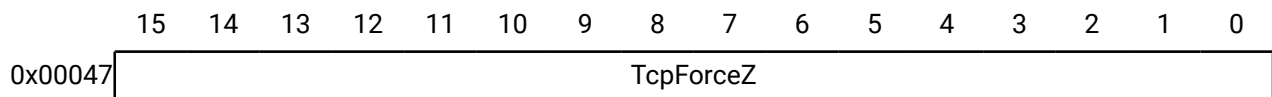
TcpForceY register type	INT16
Address	0x0046
Unit	N/100



TcpForceZ

The register holds the Z component of the force used by the TCP.

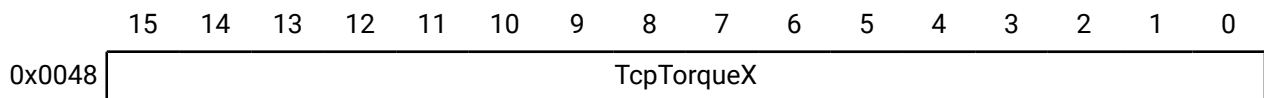
TcpForceZ register type	INT16
Address	0x0047
Unit	N/100



TcpTorqueX

The register holds the X component of the torque used by the TCP.

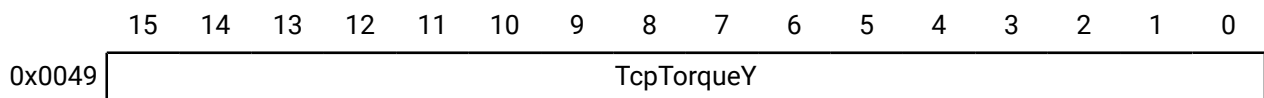
TcpTorqueX register type	INT16
Address	0x0048
Unit	N·m/100



TcpTorqueY

The register holds the Y component of the torque used by the TCP.

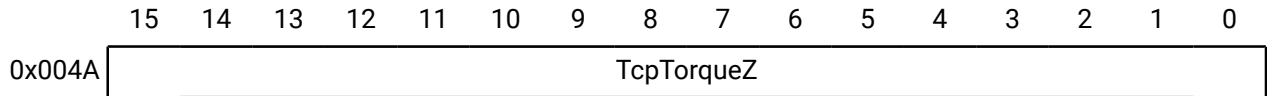
TcpTorqueY register type	INT16
Address	0x0049
Unit	N·m/100



TcpTorqueZ

The register holds the Z component of the torque used by the TCP.

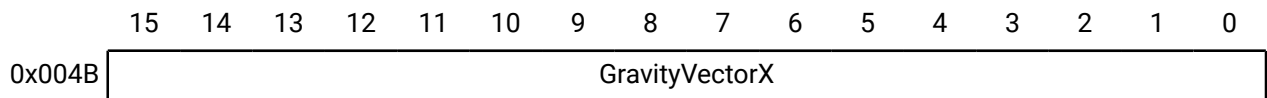
TcpTorqueZ register type	INT16
Address	0x004A
Unit	N·m/100



GravityVectorX

The register holds the X component of the gravity vector.

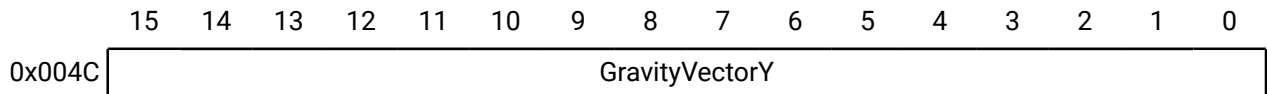
GravityVectorX register type	INT16
Address	0x004B
Unit	mm/s ²



GravityVectorY

The register holds the Y component of the gravity vector.

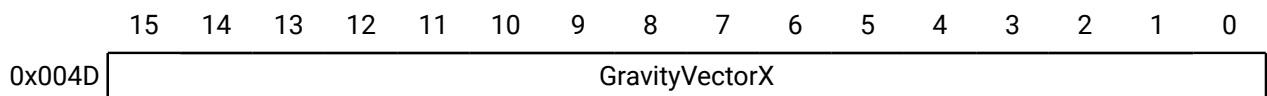
GravityVectorY register type	INT16
Address	0x004C
Unit	mm/s ²



GravityVectorZ

The register holds the Z component of the gravity vector.

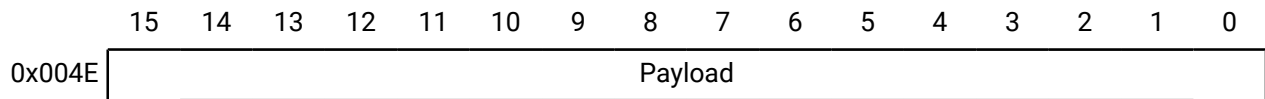
GravityVectorZ register type	INT16
Address	0x004D
Unit	mm/s ²



Payload

The register represents the mass of the tool attached to Link 6.

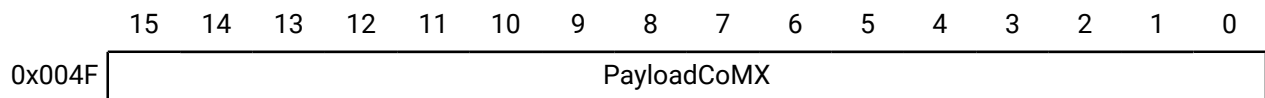
Payload register type	UINT16
Address	0x004E
Unit	g



PayloadCoMX

Similar to the `Payload` register, the register represents the X component of the center of mass of the tool attached to Link 6.

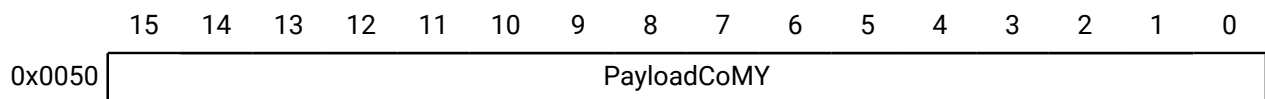
PayloadCoMX register type	INT16
Address	0x004F
Unit	mm/10



PayloadCoMY

Similar to the `Payload` register, the register represents the Y component of the center of mass of the tool attached to Link 6.

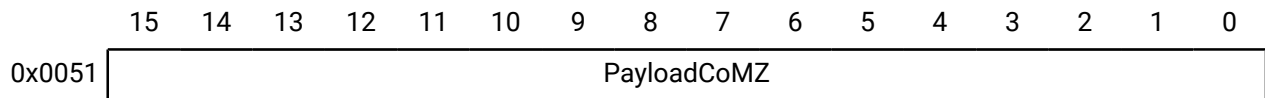
PayloadCoMY register type	INT16
Address	0x0050
Unit	mm/10



PayloadCoMZ

Similar to the `Payload` register, the register represents the Z component of the center of mass of the tool attached to Link 6.

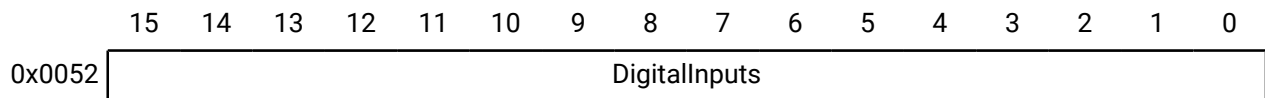
PayloadCoMZ register type	INT16
Address	0x0051
Unit	mm/10



DigitalInputs

The register represents the current state of the digital inputs on the controller.

DigitalInputs register type UINT16
 Address 0x0052
 Unit bitfield



DigitalFeedbackM8

The register is for digital feedback of the M8 connector of the wrist. If a pin is configured as an input, the associated bit contains the state of the input. If a pin is configured as an output, the associated bit contains a feedback for this output.

DigitalFeedbackM8 register type UINT16
 Address 0x0053
 Unit bitfield

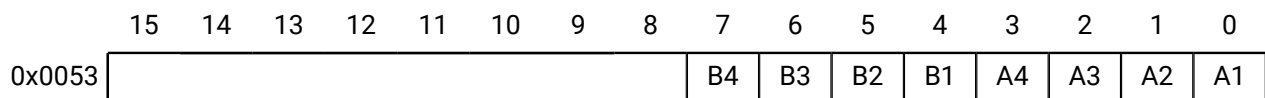


Table 9: Bits of the DigitalFeedbackM8 register

Bit	Description
B4	Actual state of the B4 pin.
B3	Actual state of the B3 pin.
B2	Actual state of the B2 pin.
B1	Actual state of the B1 pin.
A4	Actual state of the A4 pin.
A3	Actual state of the A3 pin.
A2	Actual state of the A2 pin.
A1	Actual state of the A1 pin.

AnalogInput1

The register represents the current state of the analog input on the controller.

AnalogInput1 register type UINT16
 Address 0x0054
 Unit analog

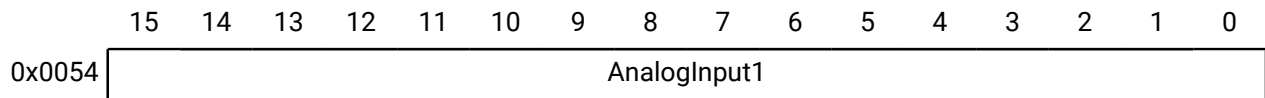


Table 10: Pin configuration and its associated analog value

Pin configuration	Analog
Voltage input	V/6553
Current input	mA/2621
Voltage output	V/744
Current output	mA/327

AnalogInput2

The register represents the current state of the analog input on the controller.

AnalogInput2 register type UINT16
 Address 0x0055
 Unit analog

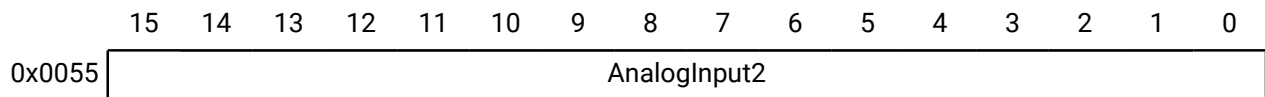


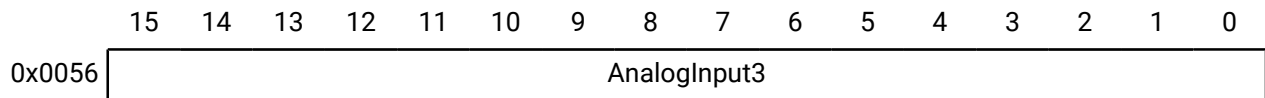
Table 11: Pin configuration and its associated analog value

Pin configuration	Analog
Voltage input	V/6553
Current input	mA/2621
Voltage output	V/744
Current output	mA/327

AnalogInput3

The register represents the current state of the analog input on the controller.

AnalogInput3 register type UINT16
 Address 0x0056
 Unit analog

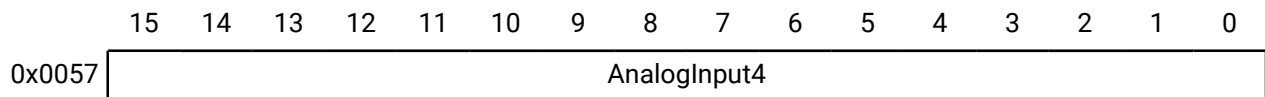
**Table 12: Pin configuration and its associated analog value**

Pin configuration	Analog
Voltage input	V/6553
Current input	mA/2621
Voltage output	V/744
Current output	mA/327

AnalogInput4

The register represents the current state of the analog input on the controller.

AnalogInput4 register type	UINT16
Address	0x0057
Unit	analog

**Table 13: Pin configuration and its associated analog value**

Pin configuration	Analog
Voltage input	V/6553
Current input	mA/2621
Voltage output	V/744
Current output	mA/327

AnalogInput1M8A

If analog pin 1 of the M8 connector A of the wrist is in VOLTAGE INPUT or CURRENT INPUT mode, and the arm is switched on, the register contains the actual voltage or current of this pin. Otherwise, the value is zero.

AnalogInput1M8A register type	UINT16
Address	0x0058
Unit	analog

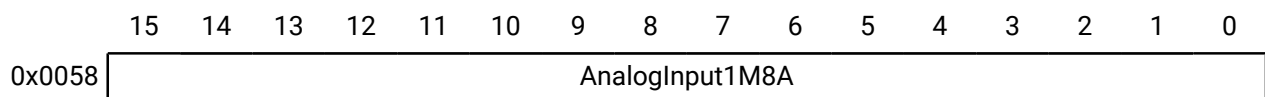


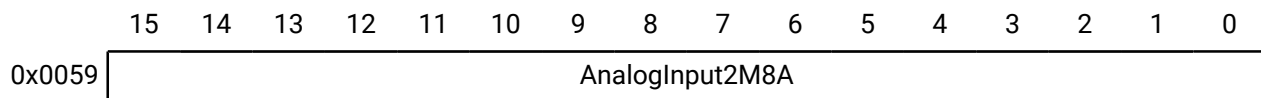
Table 14: Pin configuration and its associated analog value

Pin configuration	Analog
Voltage input	V/6553
Current input	mA/2621
Voltage output	V/744
Current output	mA/327

AnalogInput2M8A

If analog pin 2 of the M8 connector A of the wrist is in VOLTAGE INPUT or CURRENT INPUT mode, and the arm is switched on, the register contains the actual voltage or current of this pin. Otherwise, the value is zero.

AnalogInput2M8A register type UINT16
 Address 0x0059
 Unit analog

**Table 15: Pin configuration and its associated analog value**

Pin configuration	Analog
Voltage input	V/6553
Current input	mA/2621
Voltage output	V/744
Current output	mA/327

AnalogInput1M8B

For future use: There is no analog pin on connector M8B of the wrist. Its value always reads zero.

AnalogInput1M8B register type UINT16
 Address 0x005A
 Unit analog

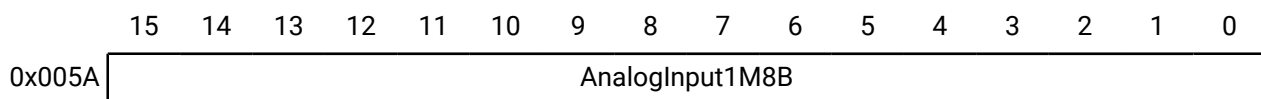


Table 16: Pin configuration and its associated analog value

Pin configuration	Analog
Voltage input	V/6553
Current input	mA/2621
Voltage output	V/744
Current output	mA/327

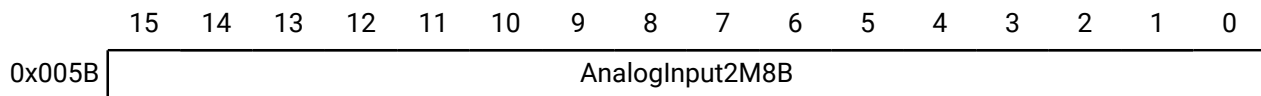
AnalogInput2M8B

For future use: There is no analog pin on connector M8B of the wrist. The value always reads zero.

AnalogInput2M8B register type UINT16

Address 0x005B

Unit analog

**Table 17: Pin configuration and its associated analog value**

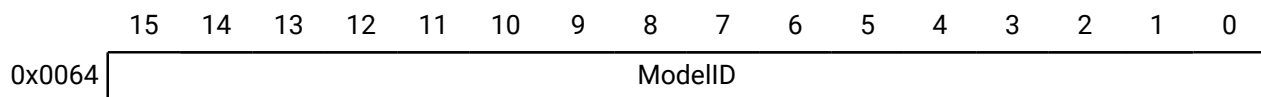
Pin configuration	Analog
Voltage input	V/6553
Current input	mA/2621
Voltage output	V/744
Current output	mA/327

ModelID

ModelID register type UINT16

Address 0x0064

Unit enumeration

**Table 18: Enumeration of the ModelID register**

Value	Description
0	MODEL_ID_UNSPECIFIED
1	MODEL_ID_L53

Value	Description
2	MODEL_ID_L31
3	MODEL_ID_HDK
4	MODEL_ID_C61

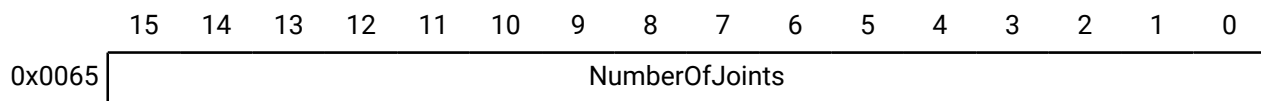
NumberOfJoints

The register holds the number of joints.

NumberOfJoints register type UINT16

Address 0x005F

Unit integer



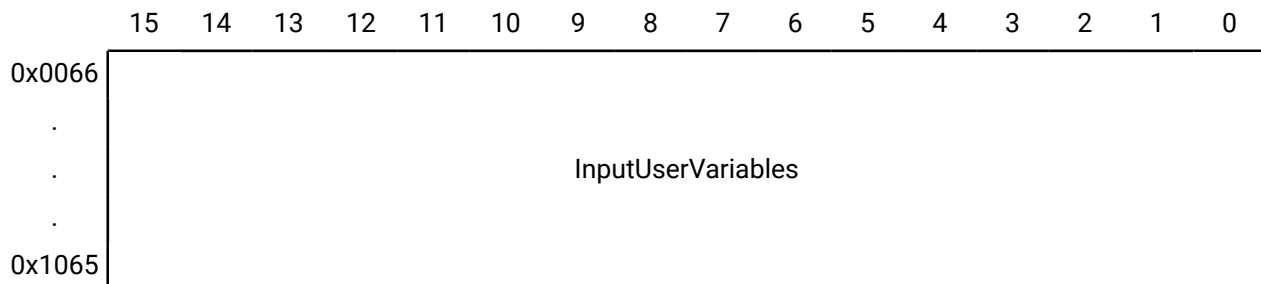
InputUserVariables

The register is configurable from the *GUI*, Kortex Web App on the **Configurations** page of the Fieldbus Adapter Plugin. Variables in this range are set by the robot and read by the PLC.

InputUserVariables register type User defined variables

Address 0x0066

Unit not applicable



Outputs detailed description

Each output register has a unique function and location in the `HOLDING_REGISTERS` address space.

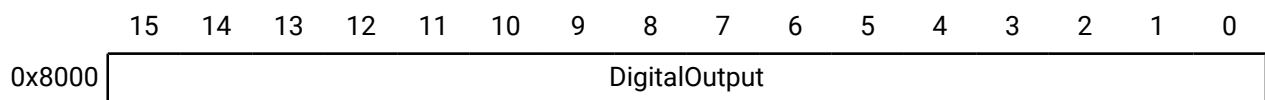
DigitalOutput

The eight least significant bits of the register, from 0 to 7, represent the current state of each of the digital outputs on the controller.

DigitalOutput register type `UINT16`

Address `0x8000`

Unit `bitfield`



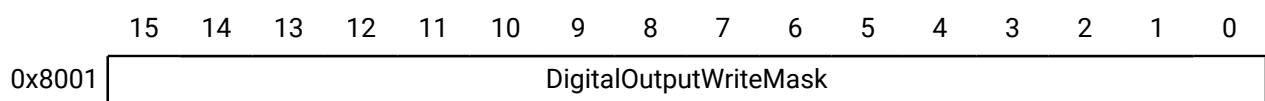
DigitalOutputWriteMask

The eight least significant bits of the register, from 0 to 7, represent which digital outputs on the controller are read-only and which are read-write. When all client connections are closed, the register is cleared.

DigitalOutputWriteMask register type `UINT16`

Address `0x8001`

Unit `bitfield`



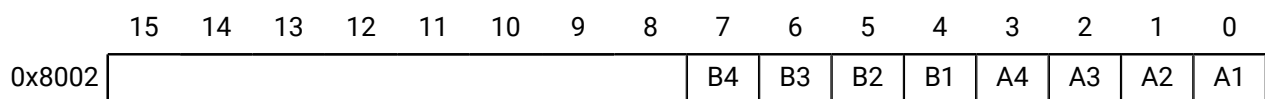
DigitalOutputM8

The eight least significant bits of the register, from 0 to 7, represent the current state of each of the digital outputs on the wrist.

DigitalOutputM8 register type `UINT16`

Address `0x8002`

Unit `bitfield`



Bit	Control pin	Signal	Description
B4	B4	WRIST_DIGITAL_IO8	- The associated bit in the <code>DigitalOutputM8WriteMask</code> must be 1 for this bit to be controlled. - The pin must also be configured as output in the user interface.
B3	B3	WRIST_DIGITAL_IO7	- The associated bit in the <code>DigitalOutputM8WriteMask</code> must be 1 for this bit to be controlled. - The pin must also be configured as output in the user interface.
B2	B2	WRIST_DIGITAL_IO6	- The associated bit in the <code>DigitalOutputM8WriteMask</code> must be 1 for this bit to be controlled. - The pin must also be configured as output in the user interface.
B1	B1	WRIST_DIGITAL_IO5	- The associated bit in the <code>DigitalOutputM8WriteMask</code> must be 1 for this bit to be controlled. - The pin must also be configured as output in the user interface.
A4	A4	WRIST_DIGITAL_IO4	- The associated bit in the <code>DigitalOutputM8WriteMask</code> must be 1 for this bit to be controlled. - The pin must also be configured as output in the user interface.
A3	A3	WRIST_DIGITAL_IO3	- The associated bit in the <code>DigitalOutputM8WriteMask</code> must be 1 for this bit to be controlled. - The pin must also be configured as output in the user interface.
A2	A2	WRIST_DIGITAL_IO2	- The associated bit in the <code>DigitalOutputM8WriteMask</code> must be 1 for this bit to be controlled. - The pin must also be configured as output in the user interface.
A1	A1	WRIST_DIGITAL_IO1	- The associated bit in the <code>DigitalOutputM8WriteMask</code> must be 1 for this bit to be controlled. - The pin must also be configured as output in the user interface.

DigitalOutputM8WriteMask

The eight least significant bits of the register, from 0 to 7, represent which digital outputs on the wrist are read-only and which are read-write. When all client connections are closed, the register is cleared.

DigitalOutputM8WriteMask register type UINT16
 Address 0x8003
 Unit bitfield

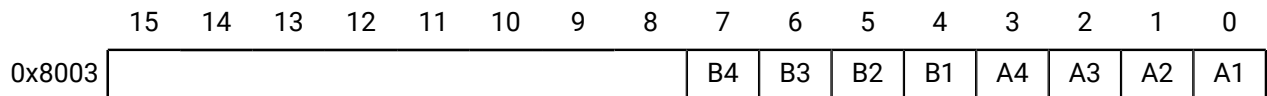


Table 19: Bitfields of the DigitalOutputM8WriteMask register

Bit	Description
B4	Control the B4 pin of the wrist.
B3	Control the B3 pin of the wrist.
B2	Control the B2 pin of the wrist.
B1	Control the B1 pin of the wrist.
A4	Control the A4 pin of the wrist.
A3	Control the A3 pin of the wrist.
A2	Control the A2 pin of the wrist.
A1	Control the A1 pin of the wrist.

AnalogOutput1

The register represents the current state of the analog output on the wrist.

AnalogOutput1 register type UINT16
 Address 0x8004
 Unit analog

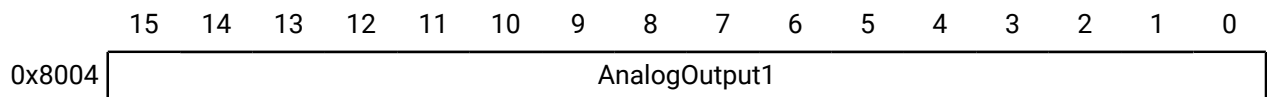


Table 20: Pin configuration and its associated analog value

Pin configuration	Analog
Voltage input	V/6553
Current input	mA/2621
Voltage output	V/744

Pin configuration	Analog
Current output	mA/327

AnalogOutput2

The register represents the current state of the analog output on the wrist.

AnalogOutput2 register type UINT16

Address 0x8005

Unit analog

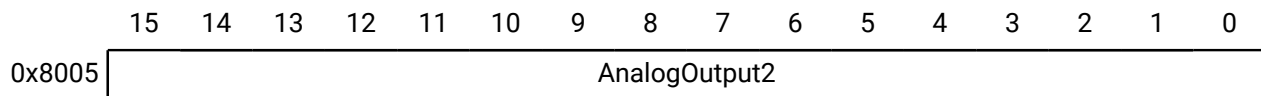


Table 21: Pin configuration and its associated analog value

Pin configuration	Analog
Voltage input	V/6553
Current input	mA/2621
Voltage output	V/744
Current output	mA/327

AnalogOutput3

The register represents the current state of the analog output on the wrist.

AnalogOutput3 register type UINT16

Address 0x8006

Unit analog

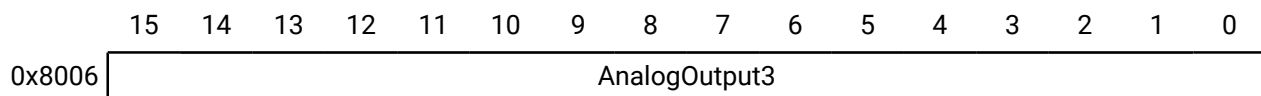


Table 22: Pin configuration and its associated analog value

Pin configuration	Analog
Voltage input	V/6553
Current input	mA/2621
Voltage output	V/744 bits
Current output	mA/327 bits

AnalogOutput4

The register represents the current state of the analog output on the wrist.

AnalogOutput4 register type	UINT16
Address	0x8007
Unit	analog

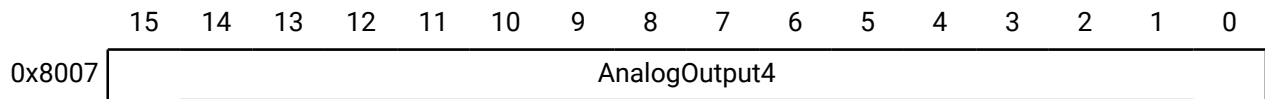


Table 23: Pin configuration and its associated analog value

Pin configuration	Analog
Voltage input	V/6553
Current input	mA/2621
Voltage output	V/744
Current output	mA/327

AnalogOutput1M8A

If analog pin 1 of the M8 connector A of the wrist is in VOLTAGE OUTPUT or CURRENT OUTPUT mode, and the arm is switched on, the register is used to command the actual voltage or current driven by this pin. Otherwise, the register does nothing.

AnalogOutput1M8A register type	UINT16
Address	0x8008
Unit	analog

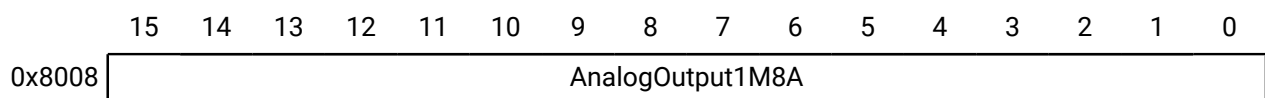


Table 24: Pin configuration and its associated analog value

Pin configuration	Analog
Voltage input	V/6553
Current input	mA/2621
Voltage output	V/744
Current output	mA/327

AnalogOutput2M8A

If analog pin 2 of the M8 connector A of the wrist is in VOLTAGE OUTPUT or CURRENT OUTPUT mode, and the robotic arm is switched on, the register is used to command the actual voltage or current driven by this pin. Otherwise, the register does nothing.

AnalogOutput2M8A register type UINT16
 Address 0x8009
 Unit analog

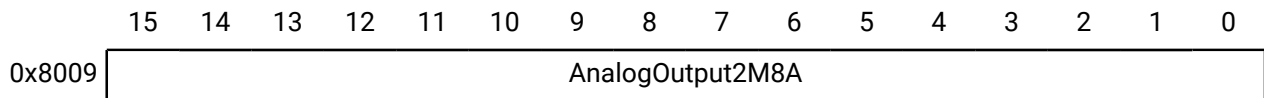


Table 25: Pin configuration and its associated analog value

Pin configuration	Analog
Voltage input	V/6553
Current input	mA/2621
Voltage output	V/744
Current output	mA/327

AnalogOutput1M8B

For future use: There is no analog pin on connector M8B of the wrist. The register does nothing.

AnalogOutput1M8B register type UINT16
 Address 0x800A
 Unit analog

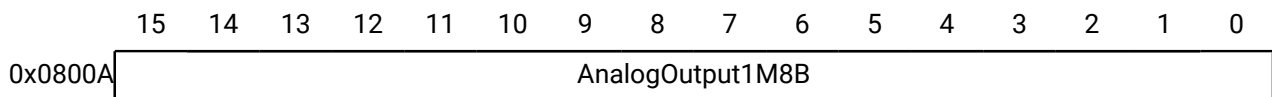


Table 26: Pin configuration and its associated analog value

Pin configuration	Analog
Voltage input	V/6553
Current input	mA/2621
Voltage output	V/744
Current output	mA/327

AnalogOutput2M8B

For future use: There is no analog pin on connector M8B of the wrist. The register does nothing.

AnalogOutput2M8B register type UINT16
 Address 0x800B
 Unit analog

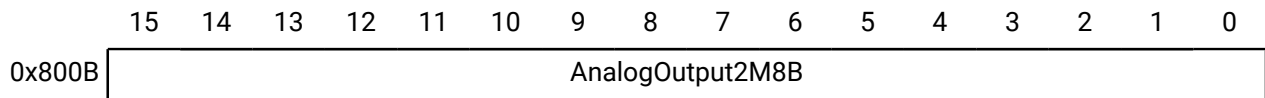


Table 27: Pin configuration and its associated analog value

Pin configuration	Analog
Voltage input	V/6553
Current input	mA/2621
Voltage output	V/744
Current output	mA/327

RobotControl

Bits 2 through 9 are used to represent the current state or operating mode of the robot. Many of the states of the robot are visible from Kortex Web App in the Robot control panel.

RobotControl register type UINT16
 Address 0x8020
 Unit bitfield

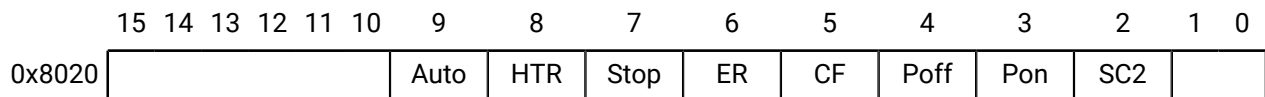


Table 28: Bitfields of the RobotControl register

Bit	Name	Description
Auto	AutomaticRising edge	Set the operating mode of the robot to AUTOMATIC.
	Poff, SC2, or Stop bits are high	Ignore the command.
	Success	The OperatingMode register is OPERATING_MODE_AUTO.
HTR	Hold-to-Run	Set the operating mode of the robot to Hold-to-Run.
	Poff, SC2, or Stop bits are high	Ignore the command.
	Success	The OperatingMode register is OPERATING_MODE_HOLD_TO_RUN.

Bit	Name	Description	
Stop	Monitored Stop	High	Hold the robotic arm in Monitored Stop mode.
		Success	The <code>OperatingMode</code> register is <code>OPERATING_MODE_MONITORED_STOP</code> .
 Important: The <code>Stop</code> bit has lower priority than the <code>Poff</code> and the <code>SC2</code> bits.			
ER	Exit Recovery	Rising edge	Change the <code>ArmState</code> register from <code>ARMSTATE_RECOVERY</code> to <code>ARMSTATE_ARM_OPERATIONAL</code> .
		<code>Poff</code> or <code>SC2</code> bits are high	Ignore the command.
		Success	The <code>ArmState</code> register is <code>ARMSTATE_ARM_OPERATIONAL</code> .
CF	Clear Faults	Rising edge	Change the <code>ArmState</code> register from <code>ARMSTATE_IN_FAULT</code> to <code>ARMSTATE_IDLE</code> .
		<code>Poff</code> or <code>SC2</code> bit are high	Ignore the command.
		Success	The <code>ArmState</code> register is <code>ARMSTATE_IDLE</code> .
Poff	Power, or switch, off	High	Hold the robotic arm in a switched off state with no power.
		Success	The <code>ArmState</code> register is <code>ARMSTATE_IDLE</code> .
 Important: The <code>Poff</code> bit has the highest level of priority within the <code>RobotControl</code> register.			
Pon	Power, or switch, on	Rising edge	Switch on the arm.
		<code>Poff</code> or <code>SC2</code> bits high	Ignore the command.
		Success	The <code>ArmState</code> register is <code>ARMSTATE_ARM_OPERATIONAL</code> .
 Important: Switching on the arm can take some time.			
SC2	Stop Category 2	High	Hold the robotic arm in a <i>Stop Category 2</i> state.
		<code>SC2</code> and <code>Poff</code> are high	The bit has no effect.
		Success	The <code>OperatingMode</code> register is <code>OPERATING_MODE_MONITORED_STOP</code> .

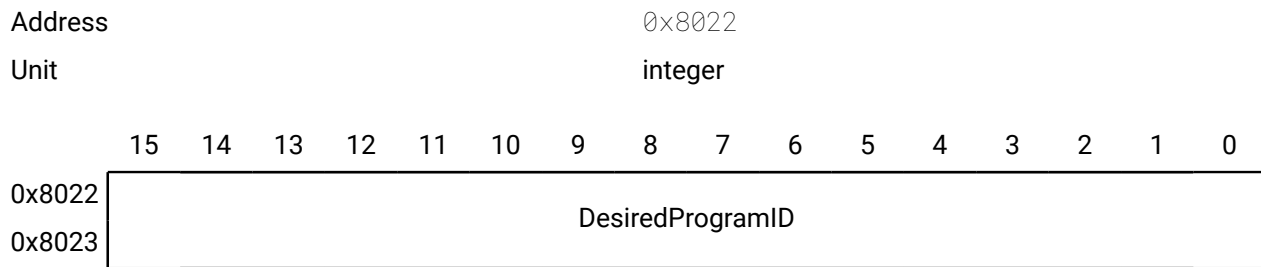
DesiredProgramID

The register latches on a rising edge of the `Start` bit of the `ProgramRunnerBits` register. In all other circumstances, the `DesiredProgramID` register is ignored by the robot.

The register holds the identifier of the program that is going to be run. The program ID is on the **Programs** page beside the program name.

`DesiredProgramID` register type

UINT32



ProgramRunnerBits

The register is used to control the running of programs with the robot.

The five least significant bits of the register, from 0 to 4, represent the current state of the program that is running.

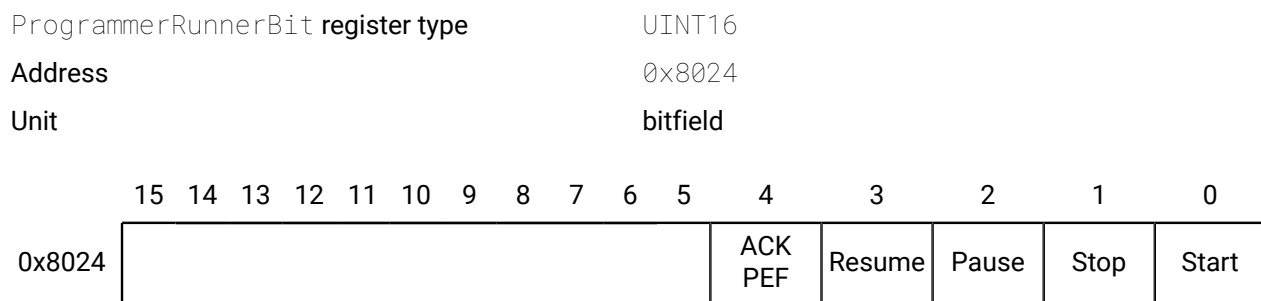


Table 29: Bits of the ProgramRunnerBits register

Bit	Description	
ACK PEF	Rising edge	Lower the PEF bit of the ProgramRunnerInputBits register.
Resume	Rising edge	Resume the playing of the current program.
	Stop or Pause bits are high	Ignore the command.
	Success	ProgramRunnerStatus register is STATUS_RUNNING.
Pause	High	Pause the playing of the current program.
		Prevent starting or resuming another program.
	Success	ProgramRunnerStatus register is STATUS_PAUSED.
Stop	High	Stop playing the current program.
		Prevent starting another program.
	Success	The ProgramRunnerStatus register is STATUS_IDLE.
Start	Rising edge	Start playing the program specified by the DesiredProgramID register.
	Stop or Pause bits are high	Ignore the command.



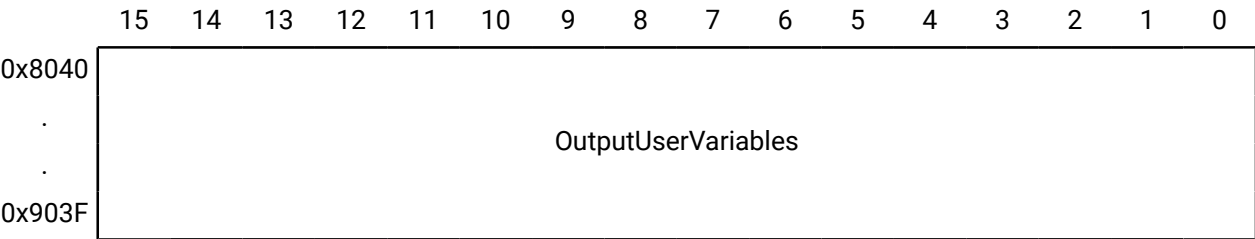
Important: The Stop bit has higher priority than Pause bit.

Bit	Description
Success	The ProgramRunnerStatus register is START_RUNNING.

OutputUserVariables

The register is configurable from Kortex Web App on the **Configurations** page of the Fieldbus Adapter Plugin. Variables in this range are set by the PLC and read by the robot.

OutputUserVariables register type	User defined variables
Address	0x8040
Unit	not applicable



Glossary

B

BE

the order of bytes in which data is stored in computer memory; the smaller memory addresses hold the most significant bytes of data and the larger memory addresses hold the least significant bytes of data

big-endian (BE)

C

controller

the part of the robot from where the software operates

CPU

the portion of the computer that retrieves and performs instructions

central processing unit (CPU)

D

DoF

the number of variables needed to define the configuration of a robot, particularly for all the directions in which the motion can occur

degrees of freedom (DoF)

E

end effector

the end of the arm of a robot; also, a tool or *flange* connected to the end of the arm of the robot

F

flange

a rim used for attaching another object

G

GUI

a way of interacting with a program of a computer that involves windows, dialogs, icons, and menus

Graphical User Interface (GUI)

synonym: human machine interface (HMI)

H

HMI

a way of interacting with a machine with such items as push buttons connected to a relay or a menu on a touch screen

human machine interface (HMI)

I

inputs

data coming from the robot and going to some form of industrial supervisory system

K

Kortex Web App

the [GUI](#) that runs on a web server on the [controller](#) and is accessed on the [teach pendant](#) or on an external computer

synonym: Web App

L

LE

the order of bytes in which data is stored in computer memory; the smaller memory addresses hold the least significant bytes of data and the larger memory addresses hold the most significant bytes of data

little-endian (LE)

M

MCU

a device used in the actuators to read sensors and report sensor information

Main Control Unit (MCU)

Modicon compatibility mode

mode of operation in which the lower registers contain the smaller bits and the higher registers contain the larger bits

MPU

a device in the controller that manages external communication and inverse kinematics, sends commands to each actuator, and collects information from all other devices

Main Processing Unit (MPU)

O

outputs

data coming from some form of industrial supervisory system to the robot

P

pendant

a device used to configure and control the movements of a robot

synonym: teach pendant

PLC

an industrial computer used in controlling manufacturing processes

programmable logic controller (PLC)

plugin

software that is added to an pre-existing base software so as to give the application additional capabilities

protective stop

a stop in which the robot stops, but there still is power in the robot

synonym: SFO2

R

RAM

the memory of the computer that stores data for the *CPU* to use

random access memory (RAM)

S

SCU

a processor in the controller that processes all safety functions

Safety Control Unit (SCU)

stop category

a type of safety control function

SC (SC)

SC₂

Stop Category 2

synonym: *protective stop*

T**teach pendant**

a device used to configure and control the movements of a robot

synonym: pendant

TCP

the position and orientation of the tool

Tool Center Point (TCP)

U**USB**

industry computer bus standard that makes it easier to transfer data between computers

Universal Serial Bus (USB)

Proudly made in North America 📍 Quebec, Canada



EN-UG-021-r1.0-202208



© 2022 Kinova inc. All rights reserved. Optional gripper shown

kinovarobotics.com
info@kinova.ca
+1 514 277-3777

KINOVA
Together in robotics