The Primary and Secondary Client Interfaces (10Hz)

The information on this page applies to:

3.4: Primary and Secondary Client Interface
3.3: Primary and Secondary Client Interface
3.2: Primary and Secondary Client Interface
3.1: Primary and Secondary Client Interface
3.0: Primary and Secondary Client Interface
1.8: Primary and Secondary Client Interface
1.7: Primary and Secondary Client Interface
1.6: Primary and Secondary Client Interface

Primary client=port 30001(Robot state and messages)

Secondary client=port 30002(Only robot state and the Version messages)

Example of data recieved: Example

The realtime communications interface (125 Hz)

The information on this page applies to UR software version

3.2 -> 3.4 Realtime interface
3.0 and 3.1 Realtime interface
pre-3.0 Realtime interface

RealTime client=port 30003(Only robot state and the Version messages)

The realtime communications interface is also known as the Matlab interface

	Updated	**3/9/2017**	**By:**	**PLN**			

Change log

Retur til index

Date	Revision	Initials	Action	Change
March 2017	3.4	RWI	Added	SW 3.3 worksheet: VersionMessage differs from SW3.2
		RWI	Added	SW 3.4 worksheet: RuntimeExceptionMessage has been changed to include the line & column number
		RWI	Added	DataStreamFromURController worksheet: Added Package-Type "Safety Data", since it was missing. Note
		PLN	Added	Change log sheet
	3.2	PLN	Corrected	ROBOT_MODE_DATA - added 8 bytes correspond to “targetSpeedFractionLimit”
	3.2	PLN	Corrected	MASTERBOARD_DATA - added 2 bytes correspond to “operationalModeSelectorInput” and “threePositi
	3.3	PLN	Corrected	In Additional Info

of the problem for the better debugging feature				
e, the package contents are not described on purpose.				
onEnablingDeviceInput”				

1.6

Retur til index

Primary and secondary client

The robot state message contain several packages as seen below.

NOTE:

- Not every robot state message may necessarily contain all of the packages described.
- Therefore, it is a good idea to test the package type chars, to see what packages are included in the message.
- Not all package types are documented here.
- Only the message type "ROBOT_STATE" is documented here (messageType value is 16)

int messageSize (total size including robot mode data, joint data, cartesian info, kinematics info, masterboard data, tool data, configuration data)

unsigned char messageType = ROBOT_STATE = 16

This page describes the data sent from the robot controller: DataStreamFromURController

Robot mode data

int packageSize

unsigned char packageType = ROBOT_MODE_DATA = 0

uint64_t timestamp

bool isRobotConnected

bool isRealRobotEnabled

bool isPowerOnRobot

bool isEmergencyStopped

bool isSecurityStopped

bool isProgramRunning

bool isProgramPaused

unsigned char robotMode

double speedFraction

Joint data

int packageSize

unsigned char packageType = JOINT_DATA = 1

for each joint:

double q_actual

double q_target

double qd_actual

float I_actual

float V_actual

float T_motor

float T_micro

unsigned char jointMode

end

The "jointMode" field is a code for the joint status (shown on the initialisation screen): JointModes

Cartesian info

int packageSize

unsigned char = CARTESIAN_INFO = 4

double X

double Y

double Z

double Rx

double Ry

double Rz

Masterboard data

int packageSize

unsigned char packageType = MASTERBOARD_DATA = 3

short digitalInputBits

short digitalOutputBits

char analogInputRange0

char analogInputRange1

double analogInput0
double analogInput1
char analogOutputDomain0
char analogOutputDomain1
double analogOutput0
double analogOutput1
float masterBoardTemperature
float robotVoltage48V
float robotCurrent
float masterIOCurrent
unsigned char masterSaftyState
unsigned char masterOnOffState
char euromap67InterfaceInstalled
(if euromap67 interface is installed, also the following:
int euromapInputBits
int euromapOutputBits
short euromapVoltage
short euromapCurrent)
Tool data
int packageSize
unsigned char packageType = TOOL_DATA = 2
char analogInputRange2
char analogInputRange3
double analogInput2
double analogInput3
float toolVoltage48V
unsigned char toolOutputVoltage
float toolCurrent
float toolTemperature
unsigned char toolMode
Below is only for primary client

The "source" field is a code for the sender of the message: MessageSources

VersionMessage - first package only

int messageSize
unsigned char messageType = ROBOT_MESSAGE = 20
uint64_t timestamp
char source
char robotMessageType = ROBOT_MESSAGE_VERSION = 3
char projectNameSize
char array projectName
unsigned char majorVersion
unsigned char minorVersion
int svnRevision
char array buildDate

Zero or more of the following messages:

SecurityMessage

int messageSize
unsigned char messageType = ROBOT_MESSAGE = 20
uint64_t timestamp
char source
char robotMessageType = ROBOT_MESSAGE_SECURITY = 5
int robotMessageCode
int robotMessageArgument
char array textMessage

RobotcommMessage

int messageSize
unsigned char messageType = ROBOT_MESSAGE = 20
uint64_t timestamp
char source
char robotMessageType = ROBOT_MESSAGE_ERROR_CODE = 6

int robotMessageCode
int robotMessageArgument
char array textMessage
KeyMessage
int messageSize
unsigned char messageType = ROBOT_MESSAGE = 20
uint64_t timestamp
char source
char robotMessageType = ROBOT_MESSAGE_KEY = 7
int robotMessageCode
int robotMessageArgument
unsigned char titleSize
char array messageTitle
char array textMessage
LabelMessage
int messageSize
unsigned char messageType = ROBOT_MESSAGE = 20
uint64_t timestamp
char source
char robotMessageType = ROBOT_MESSAGE_PROGRAM_LABEL = 1
int id
char array textMessage
PopupMessage
int messageSize
unsigned char messageType = ROBOT_MESSAGE = 20
uint64_t timestamp
char source
char robotMessageType = ROBOT_MESSAGE_POPUP = 2
bool warning
bool error

unsigned char titleSize
char array messageTitle
char array textMessage
RequestValueMessage
int messageSize
unsigned char messageType = ROBOT_MESSAGE = 20
uint64_t timestamp
char source
char robotMessageType = ROBOT_MESSAGE_REQUEST_* = 0-8
unsigned int requestId
char array textMessage
TextMessage
int messageSize
unsigned char messageType = ROBOT_MESSAGE = 20
uint64_t timestamp
char source
char robotMessageType = ROBOT_MESSAGE_TEXT = 0
char array textMessage
VarMessage
int messageSize
unsigned char messageType = ROBOT_MESSAGE = 20
uint64_t timestamp
char source
char robotMessageType = ROBOT_MESSAGE_VARIABLE = 8
int code
int argument
unsigned char titleSize
char array messageTitle
char array messageText

Example of data recieved:		Example				

1.7 Retur til index

Primary and secondary client

The robot state message contain several packages as seen below.

NOTE:

- Not every robot state message may necessarily contain all of the packages described.
- Therefore, it is a good idea to test the package type chars, to see what packages are included in the message.
- Not all package types are documented here.
- Only the message type "ROBOT_STATE" is documented here (messageType value is 16)

int messageSize (total size including robot mode data, joint data, cartesian info, kinematics info, masterboard data, tool data, configuration data)

unsigned char messageType = ROBOT_STATE = 16

This page describes the data sent from the robot controller: DataStreamFromURController

Robot mode data

int packageSize

unsigned char packageType = ROBOT_MODE_DATA = 0

uint64_t timestamp

bool isRobotConnected

bool isRealRobotEnabled

bool isPowerOnRobot

bool isEmergencyStopped

bool isSecurityStopped

bool isProgramRunning

bool isProgramPaused

unsigned char robotMode

double speedFraction

Joint data

int packageSize

unsigned char packageType = JOINT_DATA = 1

for each joint:

double q_actual

double q_target

double qd_actual

float I_actual

float V_actual

float T_motor

float T_micro

unsigned char jointMode

end

The "jointMode" field is a code for the joint status (shown on the initialisation screen): JointModes

Cartesian info

int packageSize

unsigned char = CARTESIAN_INFO = 4

double X

double Y

double Z

double Rx

double Ry

double Rz

Masterboard data

int packageSize

unsigned char packageType = MASTERBOARD_DATA = 3

short digitalInputBits

short digitalOutputBits

char analogInputRange0

char analogInputRange1

double analogInput0
double analogInput1
char analogOutputDomain0
char analogOutputDomain1
double analogOutput0
double analogOutput1
float masterBoardTemperature
float robotVoltage48V
float robotCurrent
float masterIOCurrent
unsigned char masterSaftyState
unsigned char masterOnOffState
char euromap67InterfaceInstalled
(if euromap67 interface is installed, also the following:
int euromapInputBits
int euromapOutputBits
short euromapVoltage
short euromapCurrent)
Tool data
int packageSize
unsigned char packageType = TOOL_DATA = 2
char analogInputRange2
char analogInputRange3
double analogInput2
double analogInput3
float toolVoltage48V
unsigned char toolOutputVoltage
float toolCurrent
float toolTemperature
unsigned char toolMode
Below is only for primary client

The "source" field is a code for the sender of the message: MessageSources

VesionMessage - first package only

int messageSize
unsigned char messageType = ROBOT_MESSAGE = 20
uint64_t timestamp
char source
char robotMessageType = ROBOT_MESSAGE_VERSION = 3
char projectNameSize
char array projectName
unsigned char majorVersion
unsigned char minorVersion
int svnRevision
char array buildDate

Zero or more of the following messages:

SecurityMessage

int messageSize
unsigned char messageType = ROBOT_MESSAGE = 20
uint64_t timestamp
char source
char robotMessageType = ROBOT_MESSAGE_SECURITY = 5
int robotMessageCode
int robotMessageArgument
char array textMessage

RobotcommMessage

int messageSize
unsigned char messageType = ROBOT_MESSAGE = 20
uint64_t timestamp
char source
char robotMessageType = ROBOT_MESSAGE_ERROR_CODE = 6

int robotMessageCode
int robotMessageArgument
char array textMessage
KeyMessage
int messageSize
unsigned char messageType = ROBOT_MESSAGE = 20
uint64_t timestamp
char source
char robotMessageType = ROBOT_MESSAGE_KEY = 7
int robotMessageCode
int robotMessageArgument
unsigned char titleSize
char array messageTitle
char array textMessage
LabelMessage
int messageSize
unsigned char messageType = ROBOT_MESSAGE = 20
uint64_t timestamp
char source
char robotMessageType = ROBOT_MESSAGE_PROGRAM_LABEL = 1
int id
char array textMessage
PopupMessage
int messageSize
unsigned char messageType = ROBOT_MESSAGE = 20
uint64_t timestamp
char source
char robotMessageType = ROBOT_MESSAGE_POPUP = 2
bool warning
bool error

unsigned char titleSize
char array messageTitle
char array textMessage
RequestValueMessage
int messageSize
unsigned char messageType = ROBOT_MESSAGE = 20
uint64_t timestamp
char source
char robotMessageType = ROBOT_MESSAGE_REQUEST_* = 0-8
unsigned int requestId
char array textMessage
TextMessage
int messageSize
unsigned char messageType = ROBOT_MESSAGE = 20
uint64_t timestamp
char source
char robotMessageType = ROBOT_MESSAGE_TEXT = 0
char array textMessage
VarMessage
int messageSize
unsigned char messageType = ROBOT_MESSAGE = 20
uint64_t timestamp
char source
char robotMessageType = ROBOT_MESSAGE_VARIABLE = 8
int code
int argument
unsigned char titleSize
char array messageTitle
char array messageText

Example of data recieved:		Example					

1.8 Retur til index

Primary and secondary client

The robot state message contain several packages as seen below.

NOTE:

- Not every robot state message may necessarily contain all of the packages described.
- Therefore, it is a good idea to test the package type chars, to see what packages are included in the message.
- Not all package types are documented here.
- Only the message type "ROBOT_STATE" is documented here (messageType value is 16)

int messageSize (total size including robot mode data, joint data, cartesian info, kinematics info, masterboard data, tool data, configuration data)

unsigned char messageType = ROBOT_STATE = 16

This page describes the data sent from the robot controller: DataStreamFromURController

Robot mode data

int packageSize

unsigned char packageType = ROBOT_MODE_DATA = 0

uint64_t timestamp

bool isRobotConnected

bool isRealRobotEnabled

bool isPowerOnRobot

bool isEmergencyStopped

bool isSecurityStopped

bool isProgramRunning

bool isProgramPaused

unsigned char robotMode

double speedFraction

Joint data

int packageSize

unsigned char packageType = JOINT_DATA = 1

for each joint:

double q_actual

double q_target

double qd_actual

float I_actual

float V_actual

float T_motor

float T_micro

unsigned char jointMode

end

The "jointMode" field is a code for the joint status (shown on the initialisation screen): JointModes

Cartesian info

int packageSize

unsigned char = CARTESIAN_INFO = 4

double X

double Y

double Z

double Rx

double Ry

double Rz

Masterboard data

int packageSize

unsigned char packageType = MASTERBOARD_DATA = 3

short digitalInputBits

short digitalOutputBits

char analogInputRange0

char analogInputRange1

double analogInput0
double analogInput1
char analogOutputDomain0
char analogOutputDomain1
double analogOutput0
double analogOutput1
float masterBoardTemperature
float robotVoltage48V
float robotCurrent
float masterIOCurrent
unsigned char masterSaftyState
unsigned char masterOnOffState
char euromap67InterfaceInstalled
(if euromap67 interface is installed, also the following:
int euromapInputBits
int euromapOutputBits
short euromapVoltage
short euromapCurrent)
Tool data
int packageSize
unsigned char packageType = TOOL_DATA = 2
char analogInputRange2
char analogInputRange3
double analogInput2
double analogInput3
float toolVoltage48V
unsigned char toolOutputVoltage
float toolCurrent
float toolTemperature
unsigned char toolMode
Kinematics info

int packageSize
unsigned char packageType = KINEMATICS_INFO = 5
This package contains a checksum for the specific robot. It might be subject of change in the near future
Configuration data
int packageSize
unsigned char packageType = CONFIGURATION_DATA = 6
for each joint:
double jointMinLimit
double jointMaxLimitt
for each joint:
double jointMaxSpeed
double jointMaxAcceleration
double vJointDefault
double aJointDefault
double vToolDefault
double aToolDefault
double eqRadius
for each joint:
double DHa
for each joint:
double DHd
for each joint:
double DHalpha
for each joint:
double DHtheta
int masterboardVersion
int controllerBoxType
int robotType
int robotSubType
for each joint:
int motorType

Force mode data

int packageSize

unsigned char = FORCE_MODE_DATA = 7

double X

double Y

double Z

double Rx

double Ry

double Rz

double robotDexterity

Additional info

int packageSize

unsigned char = ADDITIONAL_INFO = 8

bool teachButtonPressed

bool teachButtonEnabled

Calibration data

int packageSize

unsigned char packageType = CALIBRATION_DATA = 9

Below is only for primary client

The "source" field is a code for the sender of the message: MessageSources

VesionMessage - first package only

int messageSize

unsigned char messageType = ROBOT_MESSAGE = 20

uint64_t timestamp

char source

char robotMessageType = ROBOT_MESSAGE_VERSION = 3

char projectNameSize
char array projectName
unsigned char majorVersion
unsigned char minorVersion
int svnRevision
char array buildDate
Zero or more of the following messages:
SecurityMessage
int messageSize
unsigned char messageType = ROBOT_MESSAGE = 20
uint64_t timestamp
char source
char robotMessageType = ROBOT_MESSAGE_SECURITY = 5
int robotMessageCode
int robotMessageArgument
char array textMessage
RobotcommMessage
int messageSize
unsigned char messageType = ROBOT_MESSAGE = 20
uint64_t timestamp
char source
char robotMessageType = ROBOT_MESSAGE_ERROR_CODE = 6
int robotMessageCode
int robotMessageArgument
char array textMessage
KeyMessage
int messageSize
unsigned char messageType = ROBOT_MESSAGE = 20
uint64_t timestamp

char source
char robotMessageType = ROBOT_MESSAGE_KEY = 7
int robotMessageCode
int robotMessageArgument
unsigned char titleSize
char array messageTitle
char array textMessage
LabelMessage
int messageSize
unsigned char messageType = ROBOT_MESSAGE = 20
uint64_t timestamp
char source
char robotMessageType = ROBOT_MESSAGE_PROGRAM_LABEL = 1
int id
char array textMessage
PopupMessage
int messageSize
unsigned char messageType = ROBOT_MESSAGE = 20
uint64_t timestamp
char source
char robotMessageType = ROBOT_MESSAGE_POPUP = 2
bool warning
bool error
unsigned char titleSize
char array messageTitle
char array textMessage
RequestValueMessage
int messageSize
unsigned char messageType = ROBOT_MESSAGE = 20
uint64_t timestamp

char source
char robotMessageType = ROBOT_MESSAGE_REQUEST_* = 0-8
unsigned int requestId
char array textMessage
TextMessage
int messageSize
unsigned char messageType = ROBOT_MESSAGE = 20
uint64_t timestamp
char source
char robotMessageType = ROBOT_MESSAGE_TEXT = 0
char array textMessage
VarMessage
int messageSize
unsigned char messageType = ROBOT_MESSAGE = 20
uint64_t timestamp
char source
char robotMessageType = ROBOT_MESSAGE_VARIABLE = 8
int code
int argument
unsigned char titleSize
char array messageTitle
char array messageText
Example of data recieved: Example

3.0

Retur til index

Primary and secondary client

The robot state message contain several packages as seen below.

NOTE:

- Not every robot state message may necessarily contain all of the packages described.
- Therefore, it is a good idea to test the package type chars, to see what packages are included in the message.
- Not all package types are documented here.
- Only the message type "ROBOT_STATE" is documented here (messageType value is 16)

int messageSize (total size including robot mode data, joint data, cartesian info, kinematics info, masterboard data, tool data, configuration data)

unsigned char messageType = ROBOT_STATE = 16

This page describes the data sent from the robot controller: DataStreamFromURController

Robot mode data

int packageSize

unsigned char packageType = ROBOT_MODE_DATA = 0

uint64_t timestamp

bool isRobotConnected

bool isRealRobotEnabled

bool isPowerOnRobot

bool isEmergencyStopped

bool isProtectiveStopped

bool isProgramRunning

bool isProgramPaused

unsigned char robotMode

unsigned char controlMode

double targetSpeedFraction

double speedScaling

Joint data

int packageSize
unsigned char packageType = JOINT_DATA = 1
for each joint:
double q_actual
double q_target
double qd_actual
float I_actual
float V_actual
float T_motor
float T_micro
unsigned char jointMode
end

The "jointMode" field is a code for the joint status (shown on the initialisation screen): JointModes

Cartesian info

int packageSize
unsigned char = CARTESIAN_INFO = 4
double X
double Y
double Z
double Rx
double Ry
double Rz

Masterboard data

int packageSize
unsigned char packageType = MASTERBOARD_DATA = 3
int digitalInputBits
int digitalOutputBits
char analogInputRange0

char analogInputRange1
double analogInput0
double analogInput1
char analogOutputDomain0
char analogOutputDomain1
double analogOutput0
double analogOutput1
float masterBoardTemperature
float robotVoltage48V
float robotCurrent
float masterIOCurrent
unsigned char safetyMode
unsigned char InReducedMode
char euromap67InterfaceInstalled
(if euromap67 interface is installed, also the following:
int euromapInputBits
int euromapOutputBits
float euromapVoltage
float euromapCurrent)
uint32_t (Used by Universal Robots software only)
Tool data
int packageSize
unsigned char packageType = TOOL_DATA = 2
char analogInputRange2
char analogInputRange3
double analogInput2
double analogInput3
float toolVoltage48V
unsigned char toolOutputVoltage
float toolCurrent
float toolTemperature
unsigned char toolMode

Kinematics info

int packageSize

unsigned char packageType = KINEMATICS_INFO = 5

This package contains a checksum for the specific robot. It might be subject of change in the near future.

Configuration data

int packageSize

unsigned char packageType = CONFIGURATION_DATA = 6

for each joint:

double jointMinLimit

double jointMaxLimitt

for each joint:

double jointMaxSpeed

double jointMaxAcceleration

double vJointDefault

double aJointDefault

double vToolDefault

double aToolDefault

double eqRadius

for each joint:

double DHa

for each joint:

double DHd

for each joint:

double DHalpha

for each joint:

double DHtheta

int masterboardVersion

int controllerBoxType

int robotType

int robotSubType

Force mode data
int packageSize
unsigned char = FORCE_MODE_DATA = 7
double X
double Y
double Z
double Rx
double Ry
double Rz
double robotDexterity
Additional info
int packageSize
unsigned char = ADDITIONAL_INFO = 8
bool teachButtonPressed
bool teachButtonEnabled
Calibration data
int packageSize
unsigned char packageType = CALIBRATION_DATA = 9
This package is used internally by Universal Robots software only and should be skipped. It might be subject of change in the near future.
Below is only for primary client
The "source" field is a code for the sender of the message: MessageSources
VesionMessage - first package only
int messageSize
unsigned char messageType = ROBOT_MESSAGE = 20
uint64_t timestamp

char source	
char robotMessageType = ROBOT_MESSAGE_VERSION = 3	
char projectNameSize	
char array projectName	
unsigned char majorVersion	
unsigned char minorVersion	
int svnRevision	
char array buildDate	
Zero or more of the following messages:	
SafetyModeMessage	
int messageSize	
unsigned char messageType = ROBOT_MESSAGE = 20	
uint64_t timestamp	
char source	
char robotMessageType = ROBOT_MESSAGE_SAFETY_MODE = 5	
int robotMessageCode	
int robotMessageArgument	
char safetyModeType	
char array textMessage	
The "safetyModeType" field can contain the following values:	SafetyModeTypes
RobotcommMessage	
int messageSize	
unsigned char messageType = ROBOT_MESSAGE = 20	
uint64_t timestamp	
char source	
char robotMessageType = ROBOT_MESSAGE_ERROR_CODE = 6	
int robotMessageCode	
int robotMessageArgument	
int warningLevel	

char array textMessage	
The "warningLevel" field can contain the following values:	WarningLevels
KeyMessage	
int messageSize	
unsigned char messageType = ROBOT_MESSAGE = 20	
uint64_t timestamp	
char source	
char robotMessageType = ROBOT_MESSAGE_KEY = 7	
int robotMessageCode	
int robotMessageArgument	
unsigned char titleSize	
char array messageTitle	
char array textMessage	
LabelMessage	
int messageSize	
unsigned char messageType = ROBOT_MESSAGE = 20	
uint64_t timestamp	
char source	
char robotMessageType = ROBOT_MESSAGE_PROGRAM_LABEL = 1	
int id	
char array textMessage	
RequestValueMessage	
int messageSize	
unsigned char messageType = ROBOT_MESSAGE = 20	
uint64_t timestamp	
char source	
char robotMessageType = ROBOT_MESSAGE_REQUEST_VALUE = 9	
unsigned int requestId	
unsigned int requestedType = 0-8	

(*) bool warning
(*) bool error
(*) bool blocking
(*) unsigned char titleLength
(*) char array messageTitle
char array textMessage
(*) Fields only included in this package, if the 'requestedType' field contains the value 8. This is the 'PopupMessage' type.
TextMessage
int messageSize
unsigned char messageType = ROBOT_MESSAGE = 20
uint64_t timestamp
char source
char robotMessageType = ROBOT_MESSAGE_TEXT = 0
char array textMessage
RuntimeExceptionMessage
int messageSize
unsigned char messageType = ROBOT_MESSAGE = 20
uint64_t timestamp
char source
char robotMessageType = ROBOT_MESSAGE_RUNTIME_EXCEPTION = 10
char array textMessage
VarMessage
int messageSize
unsigned char messageType = PROGRAM_STATE_MESSAGE = 25
uint64_t timestamp
char robotMessageType = PROGRAM_STATE_MESSAGE_VARIABLE_UPDATE = 2
unsigned char titleSize
char array messageTitle
char array messageText

Deprecated:Use the much more efficient 'GlobalVariablesSetupMessage' and 'GlobalVariablesUpdateMessage' (described below).

GlobalVariablesSetupMessage

int messageSize

unsigned char messageType = PROGRAM_STATE_MESSAGE = 25

uint64_t timestamp

char robotMessageType =PROGRAM_STATE_MESSAGE_GLOBAL_VARIABLES_SETUP = 0

unsigned short startIndex

(*) char array variableNames

(*) Each new line (terminated with the '\n' character) in the array contains the name of a variable.

GlobalVariablesUpdateMessage

int messageSize

unsigned char messageType = PROGRAM_STATE_MESSAGE = 25

uint64_t timestamp

char robotMessageType = PROGRAM_STATE_MESSAGE_GLOBAL_VARIABLES_UPDATE = 1

unsigned short startIndex

for each variable: (terminated with the '\n' character)

unsigned char valueType

One of following data chunks based on the value of 'valueType':

valueType = 'NONE_VAL':

// No content -> has not been initialized yet

valueType = 'STRING_VAL':

unsigned short valueLength

char array value

valueType = 'POSE_VAL':

float x

float y

float z

float rx

float ry

float rz									
valueType = 'BOOL_VAL':									
bool value									
valueType = 'INT_VAL':									
int value									
valueType = 'FLOAT_VAL':									
float value									
valueType = 'LIST_VAL':									
unsigned short listLength									
for each item in the list:									
unsigned char valueType									
The data for that type									
The relevant values for the "valueType" field are:					ValueTypes				
The information for each variable is on a new line (terminated with the '\n' character).									
Example of data recieved:				Example					

3.1

Retur til index

Primary and secondary client

The robot state message contain several packages as seen below.

NOTE:

- Not every robot state message may necessarily contain all of the packages described.
- Therefore, it is a good idea to test the package type chars, to see what packages are included in the message.
- Not all package types are documented here.
- Only the message type "ROBOT_STATE" is documented here (messageType value is 16)

int messageSize (total size including robot mode data, joint data, cartesian info, kinematics info, masterboard data, tool data, configuration data)

unsigned char messageType = ROBOT_STATE = 16

This page describes the data sent from the robot controller: DataStreamFromURController

Robot mode data

int packageSize

unsigned char packageType = ROBOT_MODE_DATA = 0

uint64_t timestamp

bool isRobotConnected

bool isRealRobotEnabled

bool isPowerOnRobot

bool isEmergencyStopped

bool isProtectiveStopped

bool isProgramRunning

bool isProgramPaused

unsigned char robotMode

unsigned char controlMode

double targetSpeedFraction

double speedScaling

Joint data

int packageSize
unsigned char packageType = JOINT_DATA = 1
for each joint:
double q_actual
double q_target
double qd_actual
float I_actual
float V_actual
float T_motor
float T_micro
unsigned char jointMode
end

The "jointMode" field is a code for the joint status (shown on the initialisation screen): JointModes

Cartesian info

int packageSize
unsigned char = CARTESIAN_INFO = 4
double X
double Y
double Z
double Rx
double Ry
double Rz
double TCPOffsetX
double TCPOffsetY
double TCPOffsetZ
double TCPOffsetRx
double TCPOffsetRy
double TCPOffsetRz

Masterboard data
int packageSize
unsigned char packageType = MASTERBOARD_DATA = 3
int digitalInputBits
int digitalOutputBits
char analogInputRange0
char analogInputRange1
double analogInput0
double analogInput1
char analogOutputDomain0
char analogOutputDomain1
double analogOutput0
double analogOutput1
float masterBoardTemperature
float robotVoltage48V
float robotCurrent
float masterIOCurrent
unsigned char safetyMode
unsigned char InReducedMode
char euromap67InterfaceInstalled
(if euromap67 interface is installed, also the following:
int euromapInputBits
int euromapOutputBits
float euromapVoltage
float euromapCurrent)
uint32_t (Used by Universal Robots software only)
Tool data
int packageSize
unsigned char packageType = TOOL_DATA = 2
char analogInputRange2
char analogInputRange3
double analogInput2

double analogInput3
float toolVoltage48V
unsigned char toolOutputVoltage
float toolCurrent
float toolTemperature
unsigned char toolMode
Kinematics info
int packageSize
unsigned char packageType = KINEMATICS_INFO = 5
This package contains a checksum for the specific robot. It might be subject of change in the near future.
Configuration data
int packageSize
unsigned char packageType = CONFIGURATION_DATA = 6
for each joint:
double jointMinLimit
double jointMaxLimitt
for each joint:
double jointMaxSpeed
double jointMaxAcceleration
double vJointDefault
double aJointDefault
double vToolDefault
double aToolDefault
double eqRadius
for each joint:
double DHa
for each joint:
double DHd
for each joint:
double DHalpha

for each joint:

double DHtheta

int masterboardVersion

int controllerBoxType

int robotType

int robotSubType

Force mode data

int packageSize

unsigned char = FORCE_MODE_DATA = 7

double X

double Y

double Z

double Rx

double Ry

double Rz

double robotDexterity

Additional info

int packageSize

unsigned char = ADDITIONAL_INFO = 8

bool teachButtonPressed

bool teachButtonEnabled

Calibration data

int packageSize

unsigned char packageType = CALIBRATION_DATA = 9

This package is used internally by Universal Robots software only and should be skipped. It might be subject of change in the near future.

VesionMessage - first package only

This is the first package sent on both the primary and secondary client interfaces. This package it not part of the robot state message.

int messageSize

unsigned char messageType = ROBOT_MESSAGE = 20

uint64_t timestamp

char source

char robotMessageType = ROBOT_MESSAGE_VERSION = 3

char projectNameSize

char array projectName

unsigned char majorVersion

unsigned char minorVersion

int svnRevision

char array buildDate

Below is only for primary client

The "source" field is a code for the sender of the message: MessageSources

Zero or more of the following messages:

SafetyModeMessage

int messageSize

unsigned char messageType = ROBOT_MESSAGE = 20

uint64_t timestamp

char source

char robotMessageType = ROBOT_MESSAGE_SAFETY_MODE = 5

int robotMessageCode

int robotMessageArgument

char safetyModeType

char array textMessage

The "safetyModeType" field can contain the following values: SafetyModeTypes

RobotcommMessage

int messageSize
unsigned char messageType = ROBOT_MESSAGE = 20
uint64_t timestamp
char source
char robotMessageType = ROBOT_MESSAGE_ERROR_CODE = 6
int robotMessageCode
int robotMessageArgument
int warningLevel
char array textMessage
The "warningLevel" field can contain the following values: WarningLevels
KeyMessage
int messageSize
unsigned char messageType = ROBOT_MESSAGE = 20
uint64_t timestamp
char source
char robotMessageType = ROBOT_MESSAGE_KEY = 7
int robotMessageCode
int robotMessageArgument
unsigned char titleSize
char array messageTitle
char array textMessage
LabelMessage
int messageSize
unsigned char messageType = ROBOT_MESSAGE = 20
uint64_t timestamp
char source
char robotMessageType = ROBOT_MESSAGE_PROGRAM_LABEL = 1
int id
char array textMessage

RequestValueMessage

int messageSize

unsigned char messageType = ROBOT_MESSAGE = 20

uint64_t timestamp

char source

char robotMessageType = ROBOT_MESSAGE_REQUEST_VALUE = 9

unsigned int requestId

unsigned int requestedType = 0-8

(*) bool warning

(*) bool error

(*) bool blocking

(*) unsigned char titleLength

(*) char array messageTitle

char array textMessage

(*) Fields only included in this package, if the 'requestedType' field contains the value 8. This is the 'PopupMessage' type.

TextMessage

int messageSize

unsigned char messageType = ROBOT_MESSAGE = 20

uint64_t timestamp

char source

char robotMessageType = ROBOT_MESSAGE_TEXT = 0

char array textMessage

RuntimeExceptionMessage

int messageSize

unsigned char messageType = ROBOT_MESSAGE = 20

uint64_t timestamp

char source

char robotMessageType = ROBOT_MESSAGE_RUNTIME_EXCEPTION = 10

char array textMessage

VarMessage

int messageSize

unsigned char messageType = PROGRAM_STATE_MESSAGE = 25

uint64_t timestamp

char robotMessageType = PROGRAM_STATE_MESSAGE_VARIABLE_UPDATE = 2

unsigned char titleSize

char array messageTitle

char array messageText

Deprecated:Use the much more efficient 'GlobalVariablesSetupMessage' and 'GlobalVariablesUpdateMessage' (described below).

GlobalVariablesSetupMessage

int messageSize

unsigned char messageType = PROGRAM_STATE_MESSAGE = 25

uint64_t timestamp

char robotMessageType =PROGRAM_STATE_MESSAGE_GLOBAL_VARIABLES_SETUP = 0

unsigned short startIndex

(*) char array variableNames

(*) Each new line (terminated with the '\n' character) in the array contains the name of a variable.

GlobalVariablesUpdateMessage

int messageSize

unsigned char messageType = PROGRAM_STATE_MESSAGE = 25

uint64_t timestamp

char robotMessageType = PROGRAM_STATE_MESSAGE_GLOBAL_VARIABLES_UPDATE = 1

unsigned short startIndex

for each variable: (terminated with the '\n' character)

unsigned char valueType

One of following data chunks based on the value of 'valueType':

valueType = 'NONE_VAL':

// No content -> has not been initialized yet

valueType = 'STRING_VAL':

unsigned short valueLength								
char array value								
valueType = 'POSE_VAL':								
float x								
float y								
float z								
float rx								
float ry								
float rz								
valueType = 'BOOL_VAL':								
bool value								
valueType = 'INT_VAL':								
int value								
valueType = 'FLOAT_VAL':								
float value								
valueType = 'LIST_VAL':								
unsigned short listLength								
for each item in the list:								
unsigned char valueType								
The data for that type								
The relevant values for the "valueType" field are:				ValueTypes				
The information for each variable is on a new line (terminated with the '\n' character).								
Example of data recieved:			Example					

3.2

Retur til index

Primary and secondary client

The robot state message contain several packages as seen below.

NOTE:

- Not every robot state message may necessarily contain all of the packages described.
- Therefore, it is a good idea to test the package type chars, to see what packages are included in the message.
- Not all package types are documented here.
- Only the message type "ROBOT_STATE" is documented here (messageType value is 16)

int messageSize (total size including robot mode data, joint data, cartesian info, kinematics info, masterboard data, tool data, configuration data)

unsigned char messageType = ROBOT_STATE = 16

This page describes the data sent from the robot controller: DataStreamFromURController

Robot mode data

int packageSize

unsigned char packageType = ROBOT_MODE_DATA = 0

uint64_t timestamp

bool isRobotConnected

bool isRealRobotEnabled

bool isPowerOnRobot

bool isEmergencyStopped

bool isProtectiveStopped

bool isProgramRunning

bool isProgramPaused

unsigned char robotMode

unsigned char controlMode

double targetSpeedFraction

double speedScaling

double targetSpeedFractionLimit
Joint data
int packageSize
unsigned char packageType = JOINT_DATA = 1
for each joint:
double q_actual
double q_target
double qd_actual
float I_actual
float V_actual
float T_motor
float T_micro
unsigned char jointMode
end
The "jointMode" field is a code for the joint status (shown on the initialisation screen): JointModes
Cartesian info
int packageSize
unsigned char = CARTESIAN_INFO = 4
double X
double Y
double Z
double Rx
double Ry
double Rz
double TCPOffsetX
double TCPOffsetY
double TCPOffsetZ
double TCPOffsetRx
double TCPOffsetRy
double TCPOffsetRz

Masterboard data

int packageSize
unsigned char packageType = MASTERBOARD_DATA = 3
int digitalInputBits
int digitalOutputBits
char analogInputRange0
char analogInputRange1
double analogInput0
double analogInput1
char analogOutputDomain0
char analogOutputDomain1
double analogOutput0
double analogOutput1
float masterBoardTemperature
float robotVoltage48V
float robotCurrent
float masterIOCurrent
unsigned char safetyMode
unsigned char InReducedMode
char euromap67InterfaceInstalled
(if euromap67 interface is installed, also the following:
int euromapInputBits
int euromapOutputBits
float euromapVoltage
float euromapCurrent)
uint32_t (Used by Universal Robots software only)
uint8_t operationalModeSelectorInput
uint8_t threePositionEnablingDeviceInput

Tool data

int packageSize
unsigned char packageType = TOOL_DATA = 2

char analogInputRange2
char analogInputRange3
double analogInput2
double analogInput3
float toolVoltage48V
unsigned char toolOutputVoltage
float toolCurrent
float toolTemperature
unsigned char toolMode
Kinematics info
int packageSize
unsigned char packageType = KINEMATICS_INFO = 5
This package contains a checksum for the specific robot. It might be subject of change in the near future.
This information is sent when leaving initializing mode and/or if the kinematics configuration is changed.
Configuration data
int packageSize
unsigned char packageType = CONFIGURATION_DATA = 6
for each joint:
double jointMinLimit
double jointMaxLimitt
for each joint:
double jointMaxSpeed
double jointMaxAcceleration
double vJointDefault
double aJointDefault
double vToolDefault
double aToolDefault
double eqRadius
for each joint:
double DHa

for each joint:
double DHd
for each joint:
double DHalpha
for each joint:
double DHtheta
int masterboardVersion
int controllerBoxType
int robotType
int robotSubType
This information is sent when leaving initializing mode and/or if the kinematics configuration is changed.
Force mode data
int packageSize
unsigned char = FORCE_MODE_DATA = 7
unused double X
unused double Y
unused double Z
unused double Rx
unused double Ry
unused double Rz
double robotDexterity
Additional info
int packageSize
unsigned char = ADDITIONAL_INFO = 8
bool freedriveButtonPressed
bool freedriveButtonEnabled
bool IOEnabledFreedrive
Calibration data
int packageSize

unsigned char packageType = CALIBRATION_DATA = 9
This package is used internally by Universal Robots software only and should be skipped. It might be subject of change in the near future.
VersionMessage - first package only
This is the first package sent on both the primary and secondary client interfaces. This package it not part of the robot state message.
int messageSize
unsigned char messageType = ROBOT_MESSAGE = 20
uint64_t timestamp
char source
char robotMessageType = ROBOT_MESSAGE_VERSION = 3
char projectNameSize
char array projectName
unsigned char majorVersion
unsigned char minorVersion
int svnRevision
char array buildDate
Below is only for primary client
The "source" field is a code for the sender of the message: MessageSources
Zero or more of the following messages:
SafetyModeMessage
int messageSize
unsigned char messageType = ROBOT_MESSAGE = 20
uint64_t timestamp
char source
char robotMessageType = ROBOT_MESSAGE_SAFETY_MODE = 5
int robotMessageCode
int robotMessageArgument

char safetyModeType									
char array textMessage									
The "safetyModeType" field can contain the following values:					SafetyModeTypes				
RobotcommMessage									
int messageSize									
unsigned char messageType = ROBOT_MESSAGE = 20									
uint64_t timestamp									
char source									
char robotMessageType = ROBOT_MESSAGE_ERROR_CODE = 6									
int robotMessageCode									
int robotMessageArgument									
int warningLevel									
char array textMessage									
The "warningLevel" field can contain the following values:					WarningLevels				
KeyMessage									
int messageSize									
unsigned char messageType = ROBOT_MESSAGE = 20									
uint64_t timestamp									
char source									
char robotMessageType = ROBOT_MESSAGE_KEY = 7									
int robotMessageCode									
int robotMessageArgument									
unsigned char titleSize									
char array messageTitle									
char array textMessage									
LabelMessage									
int messageSize									
unsigned char messageType = ROBOT_MESSAGE = 20									

uint64_t timestamp
char source
char robotMessageType = ROBOT_MESSAGE_PROGRAM_LABEL = 1
int id
char array textMessage
RequestValueMessage
int messageSize
unsigned char messageType = ROBOT_MESSAGE = 20
uint64_t timestamp
char source
char robotMessageType = ROBOT_MESSAGE_REQUEST_VALUE = 9
unsigned int requestId
unsigned int requestedType = 0-8
(*) bool warning
(*) bool error
(*) bool blocking
(*) unsigned char titleLength
(*) char array messageTitle
char array textMessage
(*) Fields only included in this package, if the 'requestedType' field contains the value 8. This is the 'PopupMessage' type.
TextMessage
int messageSize
unsigned char messageType = ROBOT_MESSAGE = 20
uint64_t timestamp
char source
char robotMessageType = ROBOT_MESSAGE_TEXT = 0
char array textMessage
RuntimeExceptionMessage
int messageSize

unsigned char messageType = ROBOT_MESSAGE = 20

uint64_t timestamp

char source

char robotMessageType = ROBOT_MESSAGE_RUNTIME_EXCEPTION = 10

char array textMessage

VarMessage

int messageSize

unsigned char messageType = PROGRAM_STATE_MESSAGE = 25

uint64_t timestamp

char robotMessageType = PROGRAM_STATE_MESSAGE_VARIABLE_UPDATE = 2

unsigned char titleSize

char array messageTitle

char array messageText

Deprecated:Use the much more efficient 'GlobalVariablesSetupMessage' and 'GlobalVariablesUpdateMessage' (described below).

GlobalVariablesSetupMessage

int messageSize

unsigned char messageType = PROGRAM_STATE_MESSAGE = 25

uint64_t timestamp

char robotMessageType =PROGRAM_STATE_MESSAGE_GLOBAL_VARIABLES_SETUP = 0

unsigned short startIndex

(*) char array variableNames

(*) Each new line (terminated with the '\n' character) in the array contains the name of a variable.

GlobalVariablesUpdateMessage

int messageSize

unsigned char messageType = PROGRAM_STATE_MESSAGE = 25

uint64_t timestamp

char robotMessageType = PROGRAM_STATE_MESSAGE_GLOBAL_VARIABLES_UPDATE = 1

unsigned short startIndex

for each variable: (terminated with the '\n' character)
unsigned char valueType
One of following data chunks based on the value of 'valueType':
valueType = 'NONE_VAL':
// No content -> has not been initialized yet
valueType = 'STRING_VAL':
unsigned short valueLength
char array value
valueType = 'POSE_VAL':
float x
float y
float z
float rx
float ry
float rz
valueType = 'BOOL_VAL':
bool value
valueType = 'INT_VAL':
int value
valueType = 'FLOAT_VAL':
float value
valueType = 'LIST_VAL':
unsigned short listLength
for each item in the list:
unsigned char valueType
The data for that type
The relevant values for the "valueType" field are: ValueTypes
The information for each variable is on a new line (terminated with the '\n' character).
Example of data recieved: Example

3.3

Retur til index

Primary and secondary client

The following messages have been changed with respect to the previous major software release 3.2

Please consult SW3.2 documentation for all other packages.

VesionMessage - first package only

This is the first package sent on both the primary and secondary client interfaces. This package it not part of the robot state message.

int messageSize
unsigned char messageType = ROBOT_MESSAGE = 20
uint64_t timestamp
char source
char robotMessageType = ROBOT_MESSAGE_VERSION = 3
char projectNameSize
char array projectName
unsigned char majorVersion

unsigned char minorVersion										
int bugfixVersion										
int buildNumber										
char array buildDate										

3.4

Retur til index

Primary and secondary client

The following messages have been changed with respect to the previous major software release 3.3

Please consult SW3.3 documentation for all other packages.

RuntimeExceptionMessage

int messageSize

unsigned char messageType = ROBOT_MESSAGE = 20

uint64_t timestamp

char source

char robotMessageType = ROBOT_MESSAGE_RUNTIME_EXCEPTION = 10

uint32 lineNumber

uint32 columnNumber

char array textMessage

Retur til index

Secondary client communications interface

The secondary client communications interface is found at TCP port 30002.

Scheme

4 bytes (int)	Length of overall package
1 byte (uchar)	Robot MessageType
4 bytes (int)	Length of Sub-Package
1 byte (uchar)	Package-Type
n bytes	Content...
4 bytes	Length of Sub-Package
1 byte	Package-Type
n bytes	Content...
	...

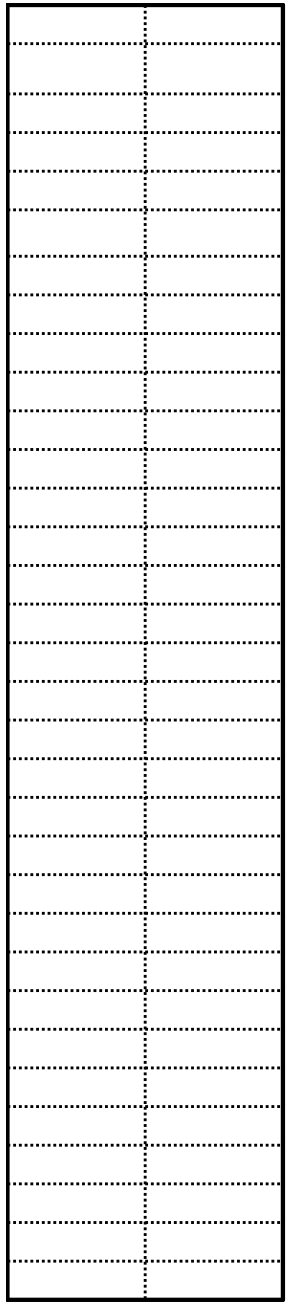
Length of overall package: total length of package including itself

Robot MessageType: only value 16 ("Robot State")

Length of Sub-Package: total length of subpackage including itself

Package-Type:

Possible types are:

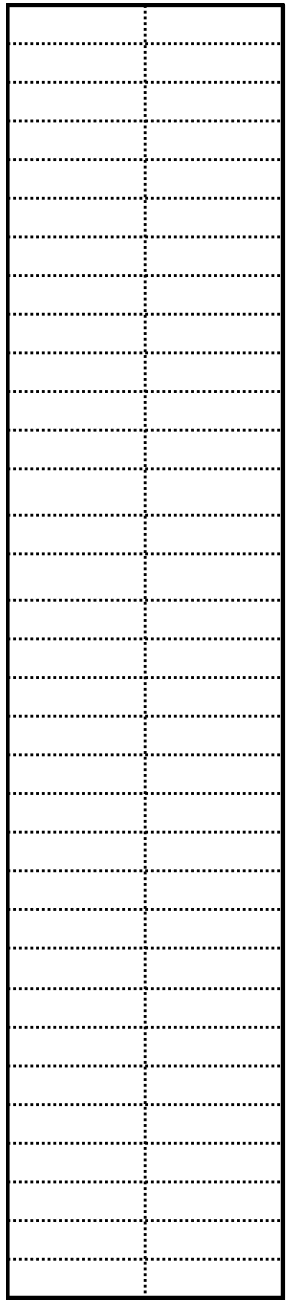


value 0 ("Robot Mode Data")
value 1 ("Joint Data")
value 2 ("Tool Data")
value 3 ("Masterboard Data")
value 4 ("Cartesian Info")
value 5 ("Kinematics Info")
value 6 ("Configuration Data")
value 7 ("Force Mode Data")
value 8 ("Additional Info")
value 9 ("Calibration Data")
value 10 ("Safety Data")

Robot State package types

Robot Mode Data (value 0)

Size in bytes	Data type	Meaning	Notes
4	integer	Package length (bytes)	46 bytes
1	uchar	Package type	Type is 0 ("Robot Mode Data")
8	uint64	Timestamp	
1	boolean	PhysicalRobotConnected	
1	boolean	RealRobotEnabled	
1	boolean	RobotPowerOn	
1	boolean	EmergencyStopped	
1	boolean	ProtectiveStopped	
1	boolean	ProgramRunning	
1	boolean	ProgramPaused	
1	uchar	Robot Mode	**See table Robot Modes**
1	uchar	ControlMode	**See table Control Modes**
8	double	TargetSpeedFraction	
8	double	SpeedScaling	
8	double	TargetSpeedFractionLimit	From 3.2



Robot Modes

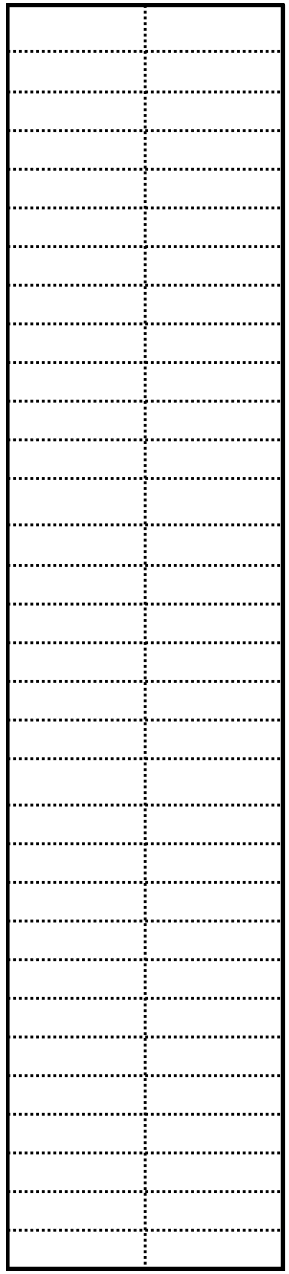
Mode	Description
0	ROBOT_MODE_DISCONNECTED
1	ROBOT_MODE_CONFIRM_SAFETY
2	ROBOT_MODE_BOOTING
3	ROBOT_MODE_POWER_OFF
4	ROBOT_MODE_POWER_ON
5	ROBOT_MODE_IDLE
6	ROBOT_MODE_BACKDRIVE
7	ROBOT_MODE_RUNNING
8	ROBOT_MODE_UPDATING_FIRMWARE

Control Modes

Mode	Description
0	CONTROL_MODE_POSITION
1	CONTROL_MODE_TEACH
2	CONTROL_MODE_FORCE
3	CONTROL_MODE_TORQUE

Joint Data (value 1)

Size in bytes	Data type	Meaning	Notes
4	integer	Package length (bytes)	251 bytes
1	uchar	Package type	Type is 1 ("Joint Data")
			All following 8 values for each joint (6x):
8	double	q actual	Actual joint position
8	double	q target	Target joint position
8	double	qd actual	Actual joint speed
4	float	I actual	Actual joint current
4	float	V actual	Actual joint voltage
4	float	T motor	Joint motor temperature
4	float	T micro	Don't use, obsolete



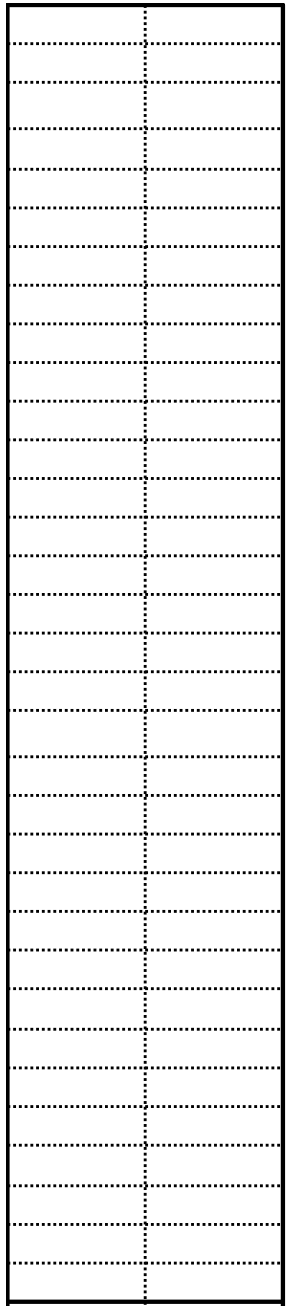
1	uchar	Joint Mode	**See table Joint Modes**

Joint Modes

Mode (uchar)	Description
236	JOINT_SHUTTING_DOWN_MODE
237	JOINT_PART_D_CALIBRATION_MODE
238	JOINT_BACKDRIVE_MODE
239	JOINT_POWER_OFF_MODE
245	JOINT_NOT_RESPONDING_MODE
246	JOINT_MOTOR_INITIALISATION_MODE
247	JOINT_BOOTING_MODE
248	JOINT_PART_D_CALIBRATION_ERROR_MODE
249	JOINT_BOOTLOADER_MODE
250	JOINT_CALIBRATION_MODE
252	JOINT_FAULT_MODE
253	JOINT_RUNNING_MODE
255	JOINT_IDLE_MODE

Tool Data (value 2)

Size in bytes	Data type	Meaning	Notes
4	integer	Package length (bytes)	37 bytes
1	uchar	Package type	Type is 2 ("Tool Data")
1	char	AnalogInputRange2	
1	char	AnalogInputRange3	
8	double	AnalogInput2	
8	double	AnalogInput3	
4	float	ToolVoltage48V	
1	uchar	ToolOutputVoltage	
4	float	ToolCurrent	
4	float	ToolTemperature	
1	uchar	Tool Mode	(A subset of joint modes)

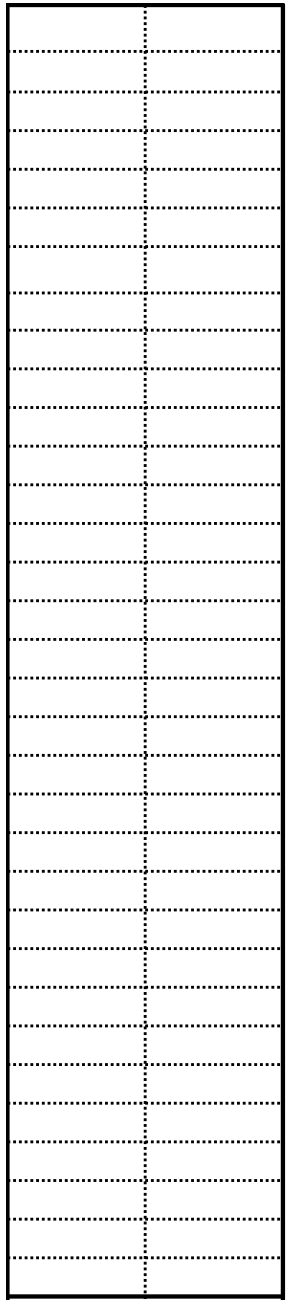


Tool Modes

Mode (uchar)	Description
249	TOOL_BOOTLOADER_MODE
253	TOOL_RUNNING_MODE
255	TOOL_IDLE_MODE

Masterboard Data (value 3)

Size in bytes	Data type	Meaning	Notes
4	integer	Package length (bytes)	74 bytes (with Euromap: 88 bytes)
1	uchar	Package type	Type is 3 ("Masterboard Data")
4	integer	DigitalInputBits?	
4	integer	DigitalOutputBits?	
1	char	AnalogInputRange0?	
1	char	AnalogInputRange1?	
8	double	AnalogInput0?	
8	double	AnalogInput1?	
1	char	AnalogOutputDomain0?	
1	char	AnalogOutputDomain? 1	
8	double	AnalogOutput0?	
8	double	AnalogOutput1?	
4	float	MasterboardTemperature?	
4	float	RobotVoltage48V?	
4	float	RobotCurrent?	
4	float	MasterIOCurrent?	
1	uchar	Safetymode	**See table SafetyMode**
1	uchar	InReducedMode?	
1	char	Euromap67Installed?	
4	integer	EuromapInputBits?	If Euromap67 is installed
4	integer	EuromapOutputBits?	If Euromap67 is installed
4	float	EuromapVoltage?	If Euromap67 is installed
4	float	EuromapCurrent?	If Euromap67 is installed
4	uint32		Used by Universal Robots software only



1	uchar	OperationalModeSelectorInput	From 3.2
1	uchar	ThreePositionEnablingDeviceInput	From 3.2

SafetyMode

State (uchar)	Description	Comment
1	SAFETY_MODE_NORMAL	
2	SAFETY_MODE_REDUCED	
3	SAFETY_MODE_PROTECTIVE_STOP	
4	SAFETY_MODE_RECOVERY	
5	SAFETY_MODE_SAFEGUARD_STOP	(SI0 + SI1 + SBUS) Physical s-stop interface input
6	SAFETY_MODE_SYSTEM_EMERGENCY_STOP	(EA + EB + SBUS->Euromap67) Physical e-stop interface input activated
7	SAFETY_MODE_ROBOT_EMERGENCY_STOP	(EA + EB + SBUS->Screen) Physical e-stop interface input activated
8	SAFETY_MODE_VIOLATION	
9	SAFETY_MODE_FAULT	

Cartesian Info (value 4)

Size in bytes	Data type	Meaning	Notes
4	integer	Package length (bytes)	101 bytes
1	uchar	Package type	type is 4 ("Cartesian Info")
8	double	X	Tool vector, X-value
8	double	Y	Tool vector, Y-value
8	double	Z	Tool vector, Z-value
8	double	Rx	Rx: Rotation vector representation of the tool orientation
8	double	Ry	Ry: Rotation vector representation of the tool orientation
8	double	Rz	Rz: Rotation vector representation of the tool orientation
8	double	TCPOffsetX?	TCP offset, X-value
8	double	TCPOffsetY?	TCP offset, Y-value
8	double	TCPOffsetZ?	TCP offset, Z-value
8	double	TCPOffsetRX?	TCP offset, Rx-value (Rotation vector representation of TCP orientatic
8	double	TCPOffsetRY?	TCP offset, Ry-value (Rotation vector representation of TCP orientatic
8	double	TCPOffsetRZ?	TCP offset, Rz-value (Rotation vector representation of TCP orientatic

on)
on)
on)

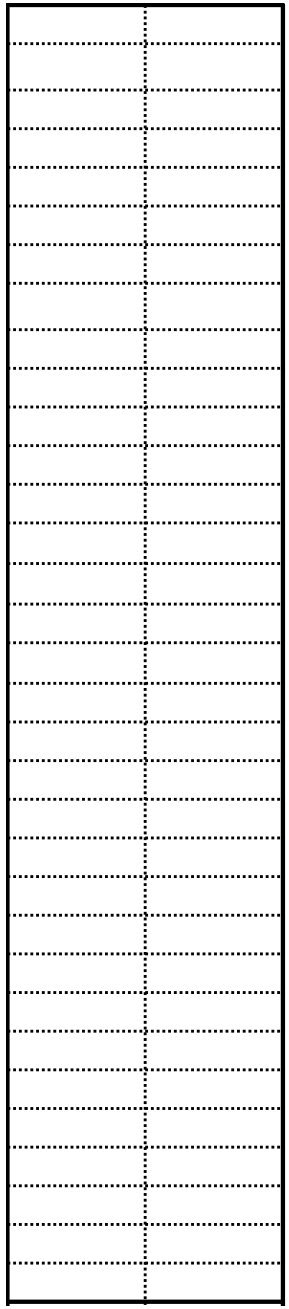
Kinematics Info (value 5)

This package has a length of 225 bytes and contains a checksum for the specific robot.

It might be subject of change in the near future.

Configuration Data (value 6)

Size in bytes	Data type	Meaning	Notes
4	integer	Package length (bytes)	445 bytes
1	uchar	Package type	type is 6 ("Configuration Data")
			2 following values for each joint (6x):
8	double	JointMinLimit?	
8	double	JointMaxLimit?	
			2 following values for each joint (6x):
8	double	JointMaxSpeed?	
8	double	JointMaxAcceleration?	
8	double	VJointDefault?	Default speed limit
8	double	AJointDefault?	Default acceleration limit
8	double	VToolDefault?	Default tool speed limit
8	double	AToolDefault?	Default tool acceleration limit
8	double	EqRadius?	The characteristic size of the tool
			1 following value for each joint (6x):
8	double	DHa	DH parameter 'a'
			1 following value for each joint (6x):
8	double	DHd	DH parameter 'd'
			1 following value for each joint (6x):
8	double	DHalpha	DH parameter 'alpha'
			1 following value for each joint (6x):
8	double	DHtheta	DH parameter 'theta'



4	integer	MasterboardVersion?	
4	integer	ControllerBoxType?	
4	integer	RobotType?	
4	integer	RobotSubType?	

Force Mode Data (value 7)

Size in bytes	Data type	Meaning	Notes
4	integer	Package length (bytes)	61 bytes
1	uchar	Package type	Type is 7 ("Force Mode Data")
8	double	X	Force mode frame, X-value
8	double	Y	Force mode frame, Y-value
8	double	Z	Force mode frame, Z-value
8	double	Rx	Force mode frame, Rx: Rotation vector
8	double	Ry	Force mode frame, Ry: Rotation vector
8	double	Rz	Force mode frame, Rz: Rotation vector
8	double	RobotDexterity?	TCP dexterity

Additional Info (value 8)

Size in bytes	Data type	Meaning	Notes
4	integer	Package length (bytes)	8 bytes
1	uchar	Package type	Type is 8 ("Additional Info")
1	boolean	FreedriveButtonPressed	
1	boolean	FreedriveButtonEnabled	
1	boolean	IOEnabledFreedrive	

Calibration Data (value 9)

This package has a length of 53 bytes. It is used internally by Universal Robots software only and should be skipped.

It might be subject of change in the near future.

Retur til index
JOINT_SHUTTING_DOWN_MODE = 236;
JOINT_PART_D_CALIBRATION_MODE = 237;
JOINT_BACKDRIVE_MODE = 238;
JOINT_POWER_OFF_MODE = 239;
JOINT_NOT_RESPONDING_MODE = 245;
JOINT_MOTOR_INITIALISATION_MODE = 246;
JOINT_BOOTING_MODE = 247;
JOINT_PART_D_CALIBRATION_ERROR_MODE = 248;
JOINT_BOOTLOADER_MODE = 249;
JOINT_CALIBRATION_MODE = 250;
JOINT_FAULT_MODE = 252;
JOINT_RUNNING_MODE = 253;
JOINT_IDLE_MODE = 255;

Retur til index

Each message sent has a "source" code for the sender of the message. The codes are:

Source is:
68: Euromap 2
67: Euromap 1
66: Teach Pendant 2
65: Teach Pendant 1
30: Safety Processor B
20: Safety Processor A
7: Controller
6: Tool
5: Wrist 3
4: Wrist 2
3: Wrist 1
2: Elbow
1: Shoulder
0: Base
-2: Robot Interface
-3: RTMachine
-4: Simulated Robot
-5: GUI

The message types are:

16: ROBOT_STATE
20: ROBOT_MESSAGE
22: HMC_MESSAGE
5: MODBUS_INFO_MESSAGE

Retur til index
SAFETY_MODE_FAULT = 9;
SAFETY_MODE_VIOLATION = 8;
SAFETY_MODE_ROBOT_EMERGENCY_STOP = 7;
SAFETY_MODE_SYSTEM_EMERGENCY_STOP = 6;
SAFETY_MODE_SAFEGUARD_STOP = 5;
SAFETY_MODE_RECOVERY = 4;
SAFETY_MODE_PROTECTIVE_STOP = 3;
SAFETY_MODE_REDUCED = 2;
SAFETY_MODE_NORMAL = 1;

Retur til index			
MESSAGE_WARNING_LEVEL_INFO = 1;			
MESSAGE_WARNING_LEVEL_WARNING = 2;			
MESSAGE_WARNING_LEVEL_VIOLATION = 3;			
MESSAGE_WARNING_LEVEL_FAULT = 4;			

Retur til index

Up to software 3.2	From software 3.3
NONE_VAL = 0;	NONE_VAL = 0,
STRING_VAL = 3;	CONST_STRING_VAL = 3,
LIST_VAL = 4;	VAR_STRING_VAL = 4,
POSE_VAL = 9;	LIST_VAL = 5,
BOOL_VAL = 11;	POSE_VAL = 10,
NUM_VAL = 12;	BOOL_VAL = 12,
INT_VAL = 13;	NUM_VAL = 13,
FLOAT_VAL = 14;	INT_VAL = 14,
	FLOAT _VAL= 15,

Retur til index

Example

Lets say the client has recieved this data:

Received 465 bytes.

Bytes: [0] [0] [1] [209] [16] [0] [0] [0] [29] [0] [0] [0] [0] [0] [0] [5] [7] [246] [1] [1] [1] [0] [0] [0] [0] [0] [63] [240] [0] [0] [0] [0] [0] [0] [0] [0]
[0] [251] [1] [64] [1] [78] [244] [77] [189] [249] [149] [64] [1] [78] [247] [89] [95] [104] [85] [0] [0] [0] [0] [0] [0] [0] [0] [188] [220] [97] [3] [66] [62]
[0] [0] [66] [0] [102] [103] [66] [99] [153] [154] [253] [191] [246] [74] [170] [216] [242] [29] [102] [191] [246] [74] [178] [44] [92] [72] [137] [0] [0] [0] [0]
[0] [0] [0] [0] [192] [1] [106] [78] [66] [63] [153] [154] [66] [4] [204] [205] [66] [104] [0] [0] [253] [63] [253] [49] [202] [91] [202] [8] [64] [63] [253] [49]
[210] [233] [51] [16] [35] [0] [0] [0] [0] [0] [0] [0] [0] [191] [155] [135] [34] [66] [62] [0] [0] [65] [249] [153] [154] [66] [100] [0] [0] [253] [191] [220] [115]
[204] [104] [205] [239] [254] [191] [220] [118] [68] [109] [49] [34] [158] [0] [0] [0] [0] [0] [0] [0] [0] [190] [60] [245] [109] [66] [63] [153] [154] [66] [25] [51]
[51] [66] [116] [204] [205] [253] [63] [242] [146] [224] [105] [231] [66] [209] [63] [242] [146] [65] [3] [193] [196] [156] [0] [0] [0] [0] [0] [0] [0] [0] [190] [115]
[157] [190] [66] [62] [0] [0] [66] [21] [51] [51] [66] [119] [153] [154] [253] [191] [231] [207] [8] [215] [85] [22] [88] [191] [231] [206] [77] [130] [151] [190] [17]
[191] [146] [242] [158] [148] [114] [240] [57] [188] [224] [224] [96] [66] [68] [102] [103] [66] [37] [153] [154] [66] [127] [153] [154] [253] [0] [0] [0] [53] [4] [63
[217] [153] [52] [224] [36] [238] [93] [191] [217] [153] [169] [67] [241] [211] [23] [63] [207] [255] [137] [8] [22] [253] [198] [63] [240] [0] [170] [111] [207] [54]
[176] [63] [243] [51] [88] [137] [58] [151] [217] [63] [201] [148] [119] [151] [70] [237] [237] [0] [0] [0] [29] [5] [64] [143] [64] [0] [0] [0] [0] [0] [64] [143] [64]
[0] [0] [0] [0] [0] [64] [143] [64] [0] [0] [0] [0] [0] [0] [0] [0] [61] [3] [0] [63] [0] [0] [0] [0] [63] [123] [129] [184] [27] [129] [184] [28] [63] [112] [225]
[14] [16] [225] [14] [17] [0] [0] [0] [0] [0] [0] [0] [0] [0] [0] [0] [0] [0] [0] [0] [0] [0] [0] [0] [0] [0] [66] [87] [51] [51] [66] [66] [0] [0] [62] [35] [215] [11] [61]
[241] [169] [253] [0] [0] [0] [37] [2] [0] [0] [63] [141] [83] [47] [180] [171] [196] [232] [63] [137] [46] [99] [102] [69] [149] [155] [66] [55] [51] [51] [0] [59]
[68] [155] [166] [66] [92] [0] [0] [253] [0]

So to start from the beginning:

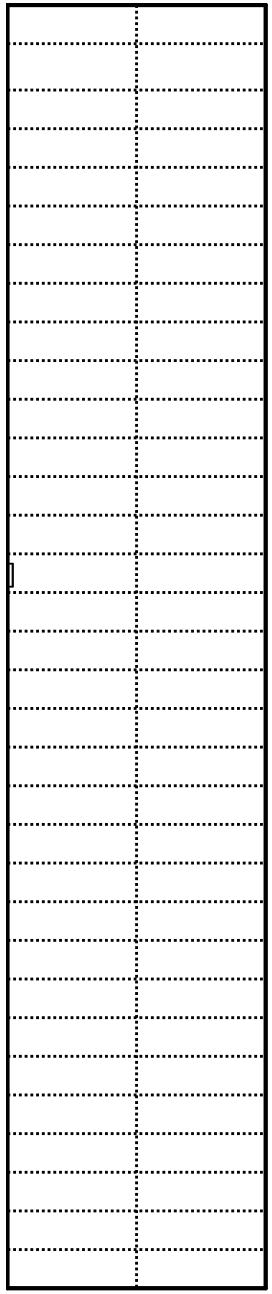
[0],[0],[1],[209]: Length is 1*256+209 = 465

[16] Package type is "Robot State"

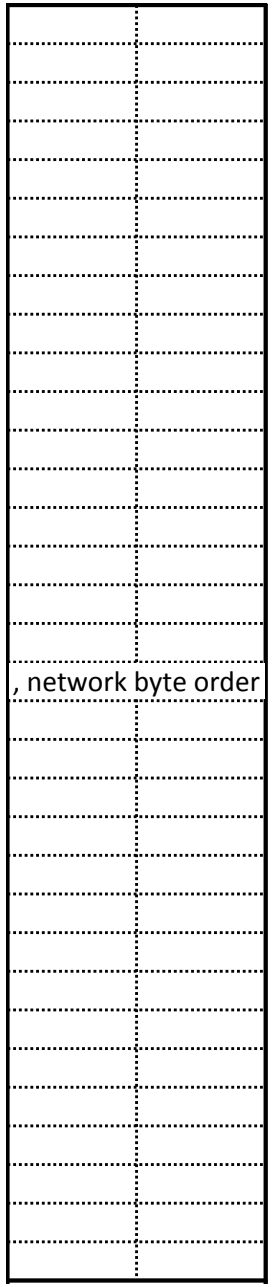
[0][0][0][29] First package length is 29

[0] Package type is 0: "Robot Mode Data"

[0] [0] [0] [0] [0] [5] [7] [246] Timestamp is 329718



[1] isRobotConnected True
[1] isRealRobotEnabled True
[1] isPowerOnRobot True
[0] isEmergencyStopped False
[0] isSecurityStopped False
[0] isProgramRunning False
[0] isProgrampaused False
[0] robot mode is 0
[63] [240] [0] [0] [0] [0] [0] [0] Speed fraction as a double see http://en.wikipedia.org/wiki/Double-precision_floating-point_format
[0] [0] [0] [251] Second package length is 251
[1] Second package type is "joint data"
[64] [1] [78][244] [77] [189] [249] [149] Joint 0 position
[64] [1] [78] [247] [89] [95] [104] [85] Joint 0 target position
[0] [0] [0] [0] [0] [0] [0] [0] Joint 0 target speed
[188] [220] [97] [3] Joint 0 current
[66] [62] [0] [0] Joint 0 voltage


, network byte order

[66] [0] [102] [103] Joint 0 motor temperature
[66] [99] [153] [154] Joint 0 microprocessor temperature
[253] Joint 0 joint mode
[191] [246] [74] [170] [216] [242] [29] [102] Joint 1 position
... etc. ...

Return to Index

Meaning	Type	Number o	Size in byt	Gnuplot c	Notes
Message Size	integer	1	4		Total message length in bytes
Time	double	1	8	1	Time elapsed since the controller was started
q target	double	6	48	2 - 7	Target joint positions
qd target	double	6	48	8 - 13	Target joint velocities
qdd target	double	6	48	14 - 19	Target joint accelerations
I target	double	6	48	20 - 25	Target joint currents
M target	double	6	48	26 - 31	Target joint moments (torques)
q actual	double	6	48	32 - 37	Actual joint positions
qd actual	double	6	48	38 - 43	Actual joint velocities
I actual	double	6	48	44 - 49	Actual joint currents
Tool Accelerometer values	double	3	24	50 - 53	Tool x,y and z accelerometer values (software version 1.7)
Unused	double	15	120	54 - 67	Unused
TCP force	double	6	48	68 - 73	Generalised forces in the TCP
Tool vector	double	6	48	74 - 79	Cartesian coordinates of the tool: (x,y,z,rx,ry,rz), where rx, ry and rz is a rotation vecto
TCP speed	double	6	48	80 - 85	Speed of the tool given in cartesian coordinates
Digital input bits	double	1	8	86	Current state of the digital inputs. NOTE: these are bits encoded as int64_t, e.g. a valu
Motor temperatures	double	6	48	87 - 92	Temperature of each joint in degrees celcius
Controller Timer	double	1	8	93	Controller realtime thread execution time
Test value	double	1	8	94	A value used by Universal Robots software only
Robot Mode	double	1	8	95	Robot control mode (see PolyScopeProgramServer on the"How to"page
Joint Modes	double	6	48	96-101	Joint control modes (see PolyScopeProgramServer on the "How to" page) (only from s
TOTAL		96	764		96 values in a 764 byte package (In software version there will be 101 values and 812 b

If it is experienced that less than 756 bytes are received, the protocol for the actual received bytes also follows the structure listed above, only not containing the e

r representation of the tool orientation
e of 5 corresponds to bit 0 and bit 2 set high
oftware version 1.8 and on)
bytes)
ntries at leading up the 756th byte.

Return to Index

Meaning	Type	Number of values	Size in bytes	Gnuplot col.	Notes
Message Size	integer	1	4		Total message length in bytes
Time	double	1	8	1	Time elapsed since the controller was started
q target	double	6	48	2 - 7	Target joint positions
qd target	double	6	48	8 - 13	Target joint velocities
qdd target	double	6	48	14 - 19	Target joint accelerations
I target	double	6	48	20 - 25	Target joint currents
M target	double	6	48	26 - 31	Target joint moments (torques)
q actual	double	6	48	32 - 37	Actual joint positions
qd actual	double	6	48	38 - 43	Actual joint velocities
I actual	double	6	48	44 - 49	Actual joint currents
I control	double	6	48	50 - 55	Joint control currents
Tool vector actual	double	6	48	56 - 61	Actual Cartesian coordinates of the tool: (x,y,z,rx,ry,rz
TCP speed actual	double	6	48	62 - 67	Actual speed of the tool given in Cartesian coordinate
TCP force	double	6	48	68 - 73	Generalised forces in the TCP
Tool vector target	double	6	48	74 - 79	Target Cartesian coordinates of the tool: (x,y,z,rx,ry,rz
TCP speed target	double	6	48	80 - 85	Target speed of the tool given in Cartesian coordinate
Digital input bits	double	1	8	86	Current state of the digital inputs. NOTE: these are bit
Motor temperatures	double	6	48	87 - 92	Temperature of each joint in degrees celsius
Controller Timer	double	1	8	93	Controller realtime thread execution time
Test value	double	1	8	94	A value used by Universal Robots software only
Robot Mode	double	1	8	95	Robot mode see DataStreamFromURC
Joint Modes	double	6	48	96-101	Joint control modes see DataStreamFromURC
Safety Mode	double	1	8	102	Safety mode see DataStreamFromURC
	double	6	48	103 - 108	Used by Universal Robots software only
Tool Accelerometer values	double	3	24	109 - 111	Tool x,y and z accelerometer values (software version
	double	6	48	112 - 117	Used by Universal Robots software only
Speed scaling	double	1	8	118	Speed scaling of the trajectory limiter
Linear momentum norm	double	1	8	119	Norm of Cartesian linear momentum
	double	1	8	120	Used by Universal Robots software only
	double	1	8	121	Used by Universal Robots software only

), where rx, ry and rz is a rotation vector representation of the tool orientation

s

), where rx, ry and rz is a rotation vector representation of the tool orientation

s

s encoded as int64_t, e.g. a value of 5 corresponds to bit 0 and bit 2 set high

ontroller

ontroller (only from software version 1.8 and on)

ontroller

1.7)

V main	double	1	8	122	Masterboard: Main voltage
V robot	double	1	8	123	Masterboard: Robot voltage (48V)
I robot	double	1	8	124	Masterboard: Robot current
V actual	double	6	48	125 - 130	Actual joint voltages
TOTAL		**131**	**1044**		131 values in a 1044 byte package

If it is experienced that less than 1044 bytes are received, the protocol for the actual received bytes also follows the structure listed above, only not containing the

entries at	leading up	the 1044th	byte.					

Return to Index

Meaning	Type	Number of values	Size in bytes	Gnuplot col.	Notes
Message Size	integer	1	4		Total message length in bytes
Time	double	1	8	1	Time elapsed since the controller was started
q target	double	6	48	2 - 7	Target joint positions
qd target	double	6	48	8 - 13	Target joint velocities
qdd target	double	6	48	14 - 19	Target joint accelerations
I target	double	6	48	20 - 25	Target joint currents
M target	double	6	48	26 - 31	Target joint moments (torques)
q actual	double	6	48	32 - 37	Actual joint positions
qd actual	double	6	48	38 - 43	Actual joint velocities
I actual	double	6	48	44 - 49	Actual joint currents
I control	double	6	48	50 - 55	Joint control currents
Tool vector actual	double	6	48	56 - 61	Actual Cartesian coordinates of the tool: (x,y,z,rx,ry,rz
TCP speed actual	double	6	48	62 - 67	Actual speed of the tool given in Cartesian coordinate
TCP force	double	6	48	68 - 73	Generalised forces in the TCP
Tool vector target	double	6	48	74 - 79	Target Cartesian coordinates of the tool: (x,y,z,rx,ry,rz
TCP speed target	double	6	48	80 - 85	Target speed of the tool given in Cartesian coordinate
Digital input bits	double	1	8	86	Current state of the digital inputs. NOTE: these are bit
Motor temperatures	double	6	48	87 - 92	Temperature of each joint in degrees celsius
Controller Timer	double	1	8	93	Controller realtime thread execution time
Test value	double	1	8	94	A value used by Universal Robots software only
Robot Mode	double	1	8	95	Robot mode see DataStreamFromURC
Joint Modes	double	6	48	96-101	Joint control modes see DataStreamFromURC
Safety Mode	double	1	8	102	Safety mode see DataStreamFromURC
	double	6	48	103 - 108	Used by Universal Robots software only
Tool Accelerometer values	double	3	24	109 - 111	Tool x,y and z accelerometer values (software version
	double	6	48	112 - 117	Used by Universal Robots software only
Speed scaling	double	1	8	118	Speed scaling of the trajectory limiter
Linear momentum norm	double	1	8	119	Norm of Cartesian linear momentum
	double	1	8	120	Used by Universal Robots software only
	double	1	8	121	Used by Universal Robots software only

), where rx, ry and rz is a rotation vector representation of the tool orientation

s

), where rx, ry and rz is a rotation vector representation of the tool orientation

s

s encoded as int64_t, e.g. a value of 5 corresponds to bit 0 and bit 2 set high

ontroller

ontroller (only from software version 1.8 and on)

ontroller

1.7)

V main	double	1	8	122	Masterboard: Main voltage
V robot	double	1	8	123	Masterboard: Robot voltage (48V)
I robot	double	1	8	124	Masterboard: Robot current
V actual	double	6	48	125 - 130	Actual joint voltages
Digital outputs	double	1	8	131	Digital outputs
Program state	double	1	8	132	Program state
TOTAL		**133**	**1060**		133 values in a 1060 byte package

If it is experienced that less than 1060 bytes are received, the protocol for the actual received bytes also follows the structure listed above, only not containing th

e entries at leading up the 1060th byte.								