The Primary and Sec	ondar	y Clie	nt In	terfa	ices ((10H	z)			
The information on this page applies to:				<u> </u>						
3.4: Primary and Secondary Client Interface										
3.3: Primary and Secondary Client Interface										
3.2: Primary and Secondary Client Interface				<u> </u>						
3.1: Primary and Secondary Client Interface										
3.0: Primary and Secondary Client Interface				·····						
1.8: Primary and Secondary Client Interface				<u>:</u>						
1.7: Primary and Secondary Client Interface				<u> </u>						
1.6: Primary and Secondary Client Interface										
	<u> </u>			<u> </u>						
Primary client=port 30001(Robot state and mess	ages)			<u>.</u>						
	<u> </u>	<u></u>		<u>.</u>						
Secondary client=port 30002(Only robot state ar	id the Version	n messages)	ļ							
				<u>.</u>						
	<u> </u>	: :		<u>:</u> :						
Example of data recieved:			Example							
	<u> </u>			<u> </u>						
The realtime commu	nicatio	ons in	terfa	ce (1	.25 F	lZ)				
						<u>/</u>				
The information on this page applies to UR softw	iare version			 !						
			ļ	 						
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)		<u></u>						
The realtime communications interface is also kn	RealTime client=port 30003(Only robot state and the Version messages) The realtime communications interface is also known as the Matlab interface									
			<u> </u>	 : :						

ĺ	Updated	3/9/2017	By:	PLN		

Change log					Retur til ind	dex							
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 woi	SW 3.4 worksheet: RuntimeExceptionMessage has been changed to include the line & column number								
		RWI	Added	DataStream	FromURCo	ntroller wo	rksheet: Ad	ded Packag	e-Type "Saf	ety Data", si	ince it was n	nissing. Note	
		PLN	Added	Change log	sheet								
	3.2	PLN	Corrected	ROBOT_MC	DE_DATA	- added 8 by	tes corresp	ond to "tar	getSpeedFr	actionLimit"			
	3.2	PLN	Corrected	MASTERBO	1ASTERBOARD_DATA - added 2 bytes correspond to "operationalModeSelectorInput" and "threePositi								
	3.3	PLN	Corrected	In Additiona	al Info								

of the prob		}		
e, the packa	ge content	are not de	scribed on p	urpose.
onEnabling	DeviceInput	"		

	:	: :			:		-	:
Primary and seco								
Primary and second	ondary	client						
The robot state message co	ntain seve	ral package	s as seen b	elow.				
NOTE:								
 Not every robot state me 								
 Therefore, it is a good ide 	a to test th	e package t	ype chars,	to see what packa	ges are included in the message.			
 Not all package types are 	document	ed here.						
 Only the message type "R 	OBOT_STA	TE" is docu	mented he	re (messageType v	alue is 16)			
								<u></u>
int messageSize (total size i	ncluding ro	bot mode o	data, joint (data, cartesian info	, kinematics info, masterboard data	a, tool data	, configuration d	ata)
unsigned char messageType	e = ROBOT_	_STATE = 16	5		÷			
	<u> </u>							
This page describes the dat	a sent from	the robot	controller:		DataStreamFromURController			
	<u> </u>							
Robot mode data	3							
int packageSize	<u>.</u>							
unsigned char packageType	e = ROBOT_	MODE_DA	ΓA = 0					
uint64_t timestamp								
bool isRobotConnected								
bool isRealRobotEnabled								
bool isPowerOnRobot								
bool isEmergencyStopped								
bool isSecurityStopped					<u>.</u>			
bool isProgramRunning		,						· · · · · · · · · · · · · · · · · · ·
bool isProgramPaused								
unsigned char robotMode								
double speedFraction								

Joint data							
int packageSize							
unsigned char packageType	= JOINT_D	ATA = 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 co	ode for the	joint status	s (shown o	n the initialisation s	screen):	<u>Joint Modes</u>	
Cartesian info							
l'.,lc:	,						
unsigned char = CARTESIAN	_INFO = 4						
double X							
double Y							
double Z							
double Rx							
double Ry							
double Rz							
Masterboard data	3						
int packageSize							
unsigned char packageType	= MASTER	BOARD_DA	TA = 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	he 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 se	nder of the message:	MessageSources	
VersionMessage - first	package only		
int messageSize			
unsigned char messageType = ROBO	_MESSAGE = 20		
uint64_t timestamp			
char source			
char robotMessageType = ROBOT_M	ESSAGE_VERSION = 3		
char projectNameSize			
char array projectName			
unsigned char majorVersion			
unsigned char minorVersion			
int svnRevision			
char array buildDate			
Zero or more of the following messag	ges:		
SecurityMessage			
int messageSize			
unsigned char messageType = ROBO	_MESSAGE = 20		
uint64_t timestamp			
char source			
char robotMessageType = ROBOT_M	ESSAGE_SECURITY = 5		
int robotMessageCode			
int robotMessageArgument			
char array textMessage			
RobotcommMessage			
int messageSize			
unsigned char messageType = ROBO	_MESSAGE = 20		
uint64_t timestamp			
char source			
char robotMessageType = ROBOT_M	ESSAGE_ERROR_CODE	= 6	

tit robotMessageArgument har array textMessage (CeyMessage it messageSize it messageSize insigned char messageType = ROBOT_MESSAGE = 20 int64_t timestamp har source har robotMessageType = ROBOT_MESSAGE_KEY = 7 it robotMessageCode it robotMessageCode it robotMessageArgument insigned char titleSize har array messageTitle har array textMessage
har array textMessage (eyMessage) It messageSize Insigned char messageType = ROBOT_MESSAGE = 20 Int64_t timestamp Inar source IntrobotMessageType = ROBOT_MESSAGE_KEY = 7 IntrobotMessageCode IntrobotMessageArgument Insigned char titleSize Inhar array messageTitle Inhar array textMessage Inhar
KeyMessage Image: Comparison of the
nt messageSize nsigned char messageType = ROBOT_MESSAGE = 20 int64_t timestamp har source har robotMessageType = ROBOT_MESSAGE_KEY = 7 nt robotMessageCode nt robotMessageArgument nsigned char titleSize har array messageTitle har array textMessage
nt messageSize nsigned char messageType = ROBOT_MESSAGE = 20 int64_t timestamp har source har robotMessageType = ROBOT_MESSAGE_KEY = 7 nt robotMessageCode nt robotMessageArgument nsigned char titleSize har array messageTitle har array textMessage
nt messageSize nsigned char messageType = ROBOT_MESSAGE = 20 int64_t timestamp har source har robotMessageType = ROBOT_MESSAGE_KEY = 7 nt robotMessageCode nt robotMessageArgument nsigned char titleSize har array messageTitle har array textMessage
int64_t timestamp har source har robotMessageType = ROBOT_MESSAGE_KEY = 7 nt robotMessageCode nt robotMessageArgument nsigned char titleSize har array messageTitle har array textMessage
har source har robotMessageType = ROBOT_MESSAGE_KEY = 7 ht robotMessageCode nt robotMessageArgument nsigned char titleSize har array messageTitle har array textMessage
har robotMessageType = ROBOT_MESSAGE_KEY = 7 It robotMessageCode It robotMessageArgument It robotMessageArgument It robotMessageArgument It robotMessageArgument It robotMessageTitle It robotMessageTitle It robotMessageTitle It robotMessage It robotMessa
nt robotMessageCode nt robotMessageArgument nsigned char titleSize har array messageTitle har array textMessage
nt robotMessageCode nt robotMessageArgument nsigned char titleSize har array messageTitle har array textMessage
nsigned char titleSize har array messageTitle har array textMessage
har array messageTitle har array textMessage
har array textMessage
.abelMessage
nt messageSize
nsigned char messageType = ROBOT_MESSAGE = 20
int64_t timestamp
har source
har robotMessageType = ROBOT_MESSAGE_PROGRAM_LABEL = 1
nt id
har array textMessage
PopupMessage PopupMessage
nt messageSize
nsigned char messageType = ROBOT_MESSAGE = 20
int64_t timestamp
har source
har robotMessageType = ROBOT_MESSAGE_POPUP = 2
ool warning
ool error

har array textMessage RequestValueMessage It messageSize Insigned char messageType = ROBOT_MESSAGE = 20 Intid4_t timestamp I						
har array textMessage RequestValueMessage Int messageSize Insigned char messageType = ROBOT_MESSAGE = 20 Insigned char messageType = ROBOT_MESSAGE_REQUEST_* = 0-8 Insigned int requestid Inhar array textMessage Int messageSize Int messageType = ROBOT_MESSAGE_TEXT = 0 Inhar array textMessage Int messageSize Int messag	unsigned char titleSize					
har array textMessage RequestValueMessage Int messageSize Insigned char messageType = ROBOT_MESSAGE = 20 Insigned char messageType = ROBOT_MESSAGE_REQUEST_* = 0-8 Insigned int requestid Inhar array textMessage Int messageSize Int messageType = ROBOT_MESSAGE_TEXT = 0 Inhar array textMessage Int messageSize Int messag	char array messageTitle					
IntressageSize Insigned char messageType = ROBOT_MESSAGE = 20 Initin64 t timestamp IntressageType = ROBOT_MESSAGE_REQUEST * = 0-8 Insigned int requestid Inhar array textMessage IntersageSize IntressageSize IntressageSize IntressageSize IntressageType = ROBOT_MESSAGE = 20 Intrestamp IntressageSize Intressa	char array textMessage					
IntressageSize Insigned char messageType = ROBOT_MESSAGE = 20 Initin64 t timestamp IntressageType = ROBOT_MESSAGE_REQUEST * = 0-8 Insigned int requestid Inhar array textMessage IntersageSize IntressageSize IntressageSize IntressageSize IntressageType = ROBOT_MESSAGE = 20 Intrestamp IntressageSize Intressa						
IntressageSize Insigned char messageType = ROBOT_MESSAGE = 20 Initin64 t timestamp IntressageType = ROBOT_MESSAGE_REQUEST * = 0-8 Insigned int requestid Inhar array textMessage IntersageSize IntressageSize IntressageSize IntressageSize IntressageType = ROBOT_MESSAGE = 20 Intrestamp IntressageSize Intressa	RequestValueMessa	ge				
Insigned char messageType = ROBOT_MESSAGE = 20 Inint64_t timestamp Aar robotMessageType = ROBOT_MESSAGE_REQUEST * = 0-8 Insigned int requestId Aar array textMessage Insigned char messageSize Insigned char messageType = ROBOT_MESSAGE = 20 Inint64_t timestamp Aar roource Aar array textMessage Insigned char messageType = ROBOT_MESSAGE = 20 Inint64_t timestamp Aar roource Aar array textMessage Insigned char messageType = ROBOT_MESSAGE_TEXT = 0 In array textMessage Insigned char messageType = ROBOT_MESSAGE_TEXT = 0 In array textMessage Insigned char messageType = ROBOT_MESSAGE = 20 Inint64_t timestamp Aar source Insigned char messageType = ROBOT_MESSAGE = 20 Inint64_t timestamp Aar source Aar robotMessageType = ROBOT_MESSAGE_VARIABLE = 8 Int code Int argument Insigned char titleSize Aar array messageTitle	int messageSize					
har source har source har robotMessageType = ROBOT_MESSAGE_REQUEST_* = 0-8 Insigned int requestid har array textMessage CextMessage	unsigned char messageType = RO	DBOT_MESSAGE =	= 20			
har source har robotMessageType = ROBOT_MESSAGE_REQUEST_* = 0-8 insigned int requestid har array textMessage har array textMessage CextMessage	uint64_t timestamp					
Insigned int requested har array textMessage	char source					
Insigned int requested har array textMessage	char robotMessageType = ROBO	T_MESSAGE_REQ	UEST_* =	0-8		
har array textMessage If extMessages In messageSize Insigned char messageType = ROBOT_MESSAGE = 20 Init64_t timestamp har source har robotMessageType = ROBOT_MESSAGE_TEXT = 0 har array textMessage In messageSize I						
nt messageSize	char array textMessage					
nt messageSize						
nt messageSize	TextMessage					
int64_t timestamp har source har robotMessageType = ROBOT_MESSAGE_TEXT = 0 har array textMessage /arMessage int messageSize insigned char messageType = ROBOT_MESSAGE = 20 int64_t timestamp har source har robotMessageType = ROBOT_MESSAGE_VARIABLE = 8 int code int argument insigned char titleSize har array messageTitle	int messageSize					
int64_t timestamp har source har robotMessageType = ROBOT_MESSAGE_TEXT = 0 har array textMessage /arMessage int messageSize insigned char messageType = ROBOT_MESSAGE = 20 int64_t timestamp har source har robotMessageType = ROBOT_MESSAGE_VARIABLE = 8 int code int argument insigned char titleSize har array messageTitle	unsigned char messageType = RC	DBOT_MESSAGE =	= 20)	
har source har robotMessageType = ROBOT_MESSAGE_TEXT = 0 har array textMessage /arMessage nt messageSize insigned char messageType = ROBOT_MESSAGE = 20 intt64_t timestamp har source har robotMessageType = ROBOT_MESSAGE_VARIABLE = 8 nt code nt argument insigned char titleSize har array messageTitle						
har robotMessageType = ROBOT_MESSAGE_TEXT = 0 har array textMessage /arMessage int messageSize insigned char messageType = ROBOT_MESSAGE = 20 intit64_t timestamp har source har robotMessageType = ROBOT_MESSAGE_VARIABLE = 8 int code int argument insigned char titleSize har array messageTitle						
har array textMessage /arMessage Int messageSize Int messageType = ROBOT_MESSAGE = 20 Intint64_t t timestamp Intar source Intar robotMessageType = ROBOT_MESSAGE_VARIABLE = 8 Int code Int argument Insigned char titleSize Intar array messageTitle Intar array messageTitle	char robotMessageType = ROBO	T_MESSAGE_TEXT	Γ = 0			
nt messageSize Insigned char messageType = ROBOT_MESSAGE = 20 Int64_t timestamp har source har robotMessageType = ROBOT_MESSAGE_VARIABLE = 8 Int code Int argument Insigned char titleSize har array messageTitle	char array textMessage					
nt messageSize Insigned char messageType = ROBOT_MESSAGE = 20 Int64_t timestamp har source har robotMessageType = ROBOT_MESSAGE_VARIABLE = 8 Int code Int argument Insigned char titleSize har array messageTitle						
nt messageSize Insigned char messageType = ROBOT_MESSAGE = 20 Int64_t timestamp har source har robotMessageType = ROBOT_MESSAGE_VARIABLE = 8 Int code Int argument Insigned char titleSize har array messageTitle	VarMessage					
Int64_t timestamp har source har robotMessageType = ROBOT_MESSAGE_VARIABLE = 8 nt code nt argument Insigned char titleSize har array messageTitle	int messageSize					
har source har robotMessageType = ROBOT_MESSAGE_VARIABLE = 8 nt code nt argument insigned char titleSize har array messageTitle	unsigned char messageType = RO	DBOT_MESSAGE =	= 20			
har source har robotMessageType = ROBOT_MESSAGE_VARIABLE = 8 nt code nt argument insigned char titleSize har array messageTitle						
nt code nt argument nt argumen	char source					
nt code nt argument nt argumen	char robotMessageType = ROBO	T_MESSAGE_VAR	IABLE = 8			
nt argument Insigned char titleSize Insigned messageTitle	int code					
har array messageTitle	int argument					
har array messageTitle	unsigned char titleSize					
	char array messageTitle					
	char array messageText					

			•	•
Evample of data recipyed:	Evamala	:	: :	:
	Examme	:	: :	
Likampic of data recieved.	LAGITIPIC	•	1 1	•
<u> </u>		<u>*</u>		

· ·	-	:	-		:				
1.7 Retur til index Primary and sec									
Primary and sec	ondary	client							
The robot state message of	ontain seve	ral package	s as seen b	elow.					
NOTE:									
Not every robot state m									
• Therefore, it is a good in	ea to test tr	ie package i	type chars,	to see wha	at packages are included in the mes	sage.		: :	
Not all package types ar	e document	ed nere.			<u> </u>			:	
Only the message type "	KORO1_S1A	TE" IS DOCU	mented he	re (messag	geType value is 16)				
int manage Cine /total sine			doto : 0:04						
				uata, carte	sian info, kinematics info, masterbo	ard data, tool d	ata, config	iration dat	a) :
unsigned char messageTy	De = ROBOT	_SIAIE = 10) 						
This page describes the da	ta sent fron	the robot	controller		DataStreamFromURController				
Tims page describes the de	1	T the robot	controller.		<u>Datastream romoneomeroner</u>				
Robot mode dat									
int packageSize									
unsigned char packageTyp	e = ROBOT_	MODE_DA	TA = 0	}					
uint64_t timestamp									
bool isRobotConnected									
bool isRealRobotEnabled									
bool isPowerOnRobot									
bool isEmergencyStopped									
bool isSecurityStopped									
bool isProgramRunning									
bool isProgramPaused									
unsigned char robotMode									
double speedFraction									ļ

									_
Joint data									
int packageSize									
unsigned char packageType	e = JOINT_D	ATA = 1							
for each joint:								<u> </u>	
double q_actual								<u> </u>	
double q_target									
double qd_actual									
float I_actual								<u> </u>	
float V_actual									
float T_motor									
float T_micro								<u>.</u>	
unsigned char jointMode							: : :	<u>.</u>	: : :
end									
The "jointMode" field is a c	ode for the	joint status	s (shown o	n the initial	lisation screen):	Joint Modes			
								<u> </u>	
Cartesian info									
int packageSize									
unsigned char = CARTESIAN	I_INFO = 4								:
double X									
double Y									
double Z									
double Rx									
double Ry									
double Rz									
Masterboard data	a								
int packageSize								•	
unsigned char packageType	e = MASTER	BOARD_DA	TA = 3					*······	!
short digitalInputBits									
short digitalOutputBits								:	:
char analogInputRange0	¢							•	
char analogInputRange1								•	!
							-	-	-

						•	
double analogInput0							
double analogInput1							
char analogOutputDomain0							
char analogOutputDomain1							
double analogOutput0							
double analogOutput1							
float masterBoardTemperat	ure						
float robotVoltage48V							
float robotCurrent							
float masterIOCurrent							
unsigned char masterSaftyS	tate						
unsigned char masterOnOffs	State						
char euromap67InterfaceIns	stalled						
(if euromap67 interface is in	stalled, als	o the follo	wing:				
int euromapInputBits							
int euromapOutputBits							
short euromapVoltage							
short euromapCurrent)							
Tool data							
int packageSize							
unsigned char packageType	= TOOL_D	4TA = 2					
char analogInputRange2							
char analogInputRange3							
double analogInput2							
double analogInput3							
float toolVoltage48V							
unsigned char toolOutputVo	ltage						
float toolCurrent							
float toolTemperature							
unsigned char toolMode							
Below is only for	primar	y clien	ıt				
	•	*		 !	:		

The "source" field is a code for the sen	der of the message:	<u>MessageSources</u>		
VesionMessage - first p	ackage only			
int massagaSiza				
unsigned char messageType = ROBOT_	MESSAGE = 20			
uint64_t timestamp				
char source				
char robotMessageType = ROBOT_ME	SSAGE_VERSION = 3			
char projectNameSize				
char array projectName				
unsigned char majorVersion				
unsigned char minorVersion				
int svnRevision				
char array buildDate				
Zero or more of the following message	es:			
SecurityMessage				
int messageSize				
unsigned char messageType = ROBOT_	MESSAGE = 20			
uint64_t timestamp				
char source				
char robotMessageType = ROBOT_ME	SSAGE_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_ME	SSAGE ERROR CODE = (5		 !

int valuat Massaca Coda	: : :	-:	Ī	;	;	;	: 1
int robotMessageCode							
int robotMessageArgument							
char array textMessage							
						ļ	
KeyMessage							
int messageSize						<u></u>	
unsigned char messageType = ROBOT	_MESSAGE = 20						
uint64_t timestamp							
char source						,	
char robotMessageType = ROBOT_ME	ESSAGE_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_ME	ESSAGE_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_ME	ESSAGE_POPUP = 2)				:	
bool warning							
bool error							
· · · · · · · · · · · · · · · · · · ·							

unsigned char titleSize								
char array messageTitle								
char array textMessage								
RequestValueMes	sage							
int messageSize								
unsigned char messageType	= ROBOT_	MESSAGE =	= 20					
uint64_t timestamp								
char source								
char robotMessageType = R	OBOT_MES	SAGE_REC	UEST_* =	0-8				
unsigned int requestId								
char array textMessage								
TextMessage								
int messageSize								
unsigned char messageType	= ROBOT_	MESSAGE =	= 20				•	
uint64_t timestamp								
char source								
char robotMessageType = Ro	OBOT_MES	SAGE_TEX	T = 0					
char array textMessage								
VarMessage								
int messageSize								
unsigned char messageType	= ROBOT_	MESSAGE =	= 20					
uint64_t timestamp							Y	
char source								
char robotMessageType = Ro	OBOT_MES	SAGE_VAR	RIABLE = 8					
int code								
int argument								
unsigned char titleSize								
char array messageTitle								
char array messageText								
					-			

Example of data recieved:	<u>Example</u>				
•		•	•		

	•					:	:		:	:
1.8	Retur til index ry and seco									
Prima	ry and seco	ndary	client							
		, , , , , , , , , , , , , , , , , , ,							<u> </u>	
The robo	t state message cor	ntain sever	al packages	as seen be	elow.					
NOTE:										
• Not eve	ery robot state mes	sage may r	ecessarily	contain all	of the pack	kages described.				
• Therefo	ore, it is a good idea	to test the	e package t	ype chars,	to see wha	t packages are included in the mes	sage.			
	package types are o									
	e message type "R(mented he	re (messag	eТуре value is 16)				
int messa	igeSize (total size ir	ncluding rol	bot mode d	lata, joint c	lata, cartes	ian info, kinematics info, masterbo	ard data, tool d	ata, config	uration dat	a)
	char messageType									
This page	describes the data	sent from	the robot o	controller:		DataStreamFromURController				
Robot	: mode data									
int packa	geSize									
unsigned	char packageType	= ROBOT_I	MODE_DAT	A = 0					·	
	timestamp									
bool isRo	botConnected									
bool isRe	alRobotEnabled									
bool isPo	werOnRobot									
bool isEm	nergencyStopped									
bool isSe	curityStopped									
bool isPro	ogramRunning									
	ogramPaused									
unsigned	char robotMode		-							
double sp	peedFraction									

Int packageSize Insigned char packageType = JOINT_DATA = 1	- · · · · · · · · · · · · · · · · · · ·						:	:	:	:
insigned char packageType = JOINT_DATA = 1 or each joint:	Joint data									
or each joint: louble q actual	int packageSize									
or each joint: louble q actual	unsigned char packageType =	JOINT_DAT	A = 1							
louble q_target louble q_target louble q_actual loat L_actual loat L_actual loat T_motor loat T_micro loat T_micro loat I_micro loat I_	for each joint:									
louble q arget louble quactual louble quacture quactual louble quacture quactu	double q_actual									
Jost J actual Jost J actual	double q_target									
loat T motor loat T micro	double qd_actual									
loat T motor loat T micro	float I_actual									
loat T_micro loat T_micro	float V_actual									
loat T_micro	float T_motor									
The "jointMode" field is a code for the joint status (shown on the initialisation screen): Cartesian info Int packageSize Insigned char = CARTESIAN_INFO = 4 Idouble X Idouble Y Idouble Z Idouble Rx	float T_micro									
Cartesian info It packageSize Insigned char = CARTESIAN_INFO = 4 Idouble X Idouble X Idouble Z Idouble Rx Idouble Rx Idouble Ry Idouble Rz Ido	unsigned char jointMode									
Cartesian info nt packageSize unsigned char = CARTESIAN_INFO = 4 louble X louble Y louble Z louble Rx louble Ry louble Rz louble Rz louble Rz Masterboard data nt packageSize unsigned char packageType = MASTERBOARD_DATA = 3 lhort digitalInputBits thar analogInputRange0	end									
Cartesian info nt packageSize unsigned char = CARTESIAN_INFO = 4 louble X louble Y louble Z louble Rx louble Ry louble Rz louble Rz louble Rz Masterboard data nt packageSize unsigned char packageType = MASTERBOARD_DATA = 3 lhort digitalInputBits thar analogInputRange0	ļ.									
Int packageSize	The "jointMode" field is a cod	le for the joi	nt status (shown on	the initial	isation screen):	<u>Joint Modes</u>			
Int packageSize										
Insigned char = CARTESIAN_INFO = 4 Idouble X Idouble Y Idouble Z Idouble RX Idouble RX Idouble RX Idouble RZ I	Cartesian info									
Insigned char = CARTESIAN_INFO = 4 Idouble X Idouble Y Idouble Z Idouble RX Idouble RX Idouble RX Idouble RZ I	int packageSize									
double X	unsigned char = CARTESIAN_I	NFO = 4								
Souble Rx Soub	double X									
double Rx double Rz double Rz Masterboard data nt packageSize unsigned char packageType = MASTERBOARD_DATA = 3 hort digitalInputBits hort digitalOutputBits char analogInputRange0	double Y									
double Rx double Rz double Rz Masterboard data nt packageSize unsigned char packageType = MASTERBOARD_DATA = 3 hort digitalInputBits hort digitalOutputBits char analogInputRange0	double Z									
Houble Ry Houble Rz Houble	double Rx									
Masterboard data Int packageSize Int packageType = MASTERBOARD_DATA = 3 Inhort digitalInputBits Inhort digitalOutputBits Inhort digitalOutputBits Internal analogInputRangeO	double Ry									
nt packageSize unsigned char packageType = MASTERBOARD_DATA = 3 whort digitalInputBits whort digitalOutputBits char analogInputRange0	double Rz									
nt packageSize unsigned char packageType = MASTERBOARD_DATA = 3 whort digitalInputBits whort digitalOutputBits char analogInputRange0										
nt packageSize unsigned char packageType = MASTERBOARD_DATA = 3 whort digitalInputBits whort digitalOutputBits char analogInputRange0	Masterboard data									
unsigned char packageType = MASTERBOARD_DATA = 3 short digitalInputBits short digitalOutputBits short analogInputRange0										
hort digitalInputBits hort digitalOutputBits har analogInputRange0	unsigned char packageType =	MASTERBO	ARD_DATA	4 = 3						!
char analogInputRangeO	short digitalInputBits									
char analogInputRangeO									·	
	char analogInputRange0									
										!

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 euromapVoltage					
short euromapCurrent)					
Tool data					
int packageSize					
unsigned char packageType = TOOL_DA	TA = 2				
char analogInputRange2					
char analogInputRange3					
double analogInput2					
double analogInput3					
float toolVoltage48V					
unsigned char toolOutputVoltage	į				
float toolCurrent					
float toolTemperature					
unsigned char toolMode					
	į				
Kinematics info					
<u> </u>	.	:	<u>:</u>		

int packageSize								
unsigned char packageType	= KINEMAT	ICS INFO =	: = 5				 !	
l :								
This package contains a chec	ksum for t	he specific	robot. It m	imight be sub	i nject of change in the near future		<u>.</u>	
					, , , , , , , , , , , , , , , , , , , ,		 i	
Configuration dat	:a						 	
int packageSize							 :	
unsigned char packageType	– CONFIGI	IRATION D	 ΛΤΛ – 6				 : :	
for each joint:	- CONTIGO	NATION_D	A1A - 0				<u>:</u> :	
double jointMinLimit							 :	
double jointMaxLimitt								
for each joint:								
double jointMaxSpeed							 	
double jointMaxAcceleration	າ 						 <u>.</u>	
double vJointDefault								
double aJointDefault							 	
double vToolDefault							 	
double aToolDefault							<u>.</u>	
double eqRadius						<u>.</u>		
for each joint:							 <u></u>	
double DHa								
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							 ٥ : :	
· · ·				-	:	:	 	

		:				
Force mode data						i
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	3					
bool teachButtonPressed						
bool teachButtonEnabled						
Calibration data						
int packageSize						
unsigned char packageType = CALIBRA	TION_DATA = 9					
Below is only for prima	ry client					
Below is only for prima The "source" field is a code for the ser	nder of the message:		MessageSources			
VesionMessage - first p	ackage only					
int messageSize				!	•	!
unsigned char messageType = ROBOT	_MESSAGE = 20					
uint64_t timestamp						
char source						
char robotMessageType = ROBOT_ME	SSAGE_VERSION = 3					

char projectNameSize					
char array projectName					
unsigned char majorVersion					
unsigned char minorVersion					
int svnRevision					
char array buildDate					
Zero or more of the following message	es:				
SecurityMessage					
int messageSize					
unsigned char messageType = ROBOT_	_MESSAGE = 20				
uint64_t timestamp					
char source					
char robotMessageType = ROBOT_ME	SSAGE_SECURITY = !	5			
int robotMessageCode					
int robotMessageArgument					
char array textMessage					
				<u>:</u>	
RobotcommMessage					
int messageSize					
unsigned char messageType = ROBOT_	MESSAGE = 20			<u> </u>	
uint64_t timestamp					
char source					
char robotMessageType = ROBOT_ME	SSAGE_ERROR_COD	E = 6			
int robotMessageCode					
int robotMessageArgument					
char array textMessage					
KeyMessage					
int messageSize					
unsigned char messageType = ROBOT_	_MESSAGE = 20			¢	
uint64_t timestamp				•	

char robotMessageType = ROBOT_MESSAGE_KEY = 7 int robotMessageArgument unsigned char itileSize char array messageTitle char array textMessage LabeIMessage 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 messageSize unsigned char messageType = ROBOT_MESSAGE = 20 uint64_t messageSize unsigned char messageType = ROBOT_MESSAGE = 20 uint64_t messageSize unsigned char messageType = ROBOT_MESSAGE = 20 uint64_t messageType = ROBOT_MESSAGE = 20 uint64_t messageType = ROBOT_MESSAGE = 20 uint64_t messageType = ROBOT_MESSAGE_POPUP = 2 bool warning bool warning bool warning bool of titleSize char array textMessage RequestValueMessage RequestValueMessage unsigned char messageType = ROBOT_MESSAGE = 20 uint messageType = ROBOT_MESSAGE_POPUP = 2 bool warning bool wa	:						:	:
int robotMessageArgument unsigned char ititleSize char array messageTitle char array textMessage LabelMessage int messageSize unsigned char messageType = ROBOT_MESSAGE = 20 unt64_t timestamp char source char robotMessageType = ROBOT_MESSAGE_PROGRAM_LABEL = 1 int id int id char array textMessage POPUPMESSAGE Int messageSize unsigned char messageType = ROBOT_MESSAGE = 20 unt64_t timestamp char source char robotMessageType = ROBOT_MESSAGE_PROGRAM_LABEL = 1 int id int id char array textMessage POPUPMESSAGE Int messageSize unsigned char messageType = ROBOT_MESSAGE = 20 unt64_t timestamp char source char robotMessageType = ROBOT_MESSAGE_POPUP = 2 bool warning bool errori unsigned char titleSize char array textMessage REQUESTVAILUEMESSAGE unsigned char titleSize char array textMessage REQUESTVAILUEMESSAGE unsigned char messageTitle char array textMessage REQUESTVAILUEMESSAGE unsigned char messageType = ROBOT_MESSAGE = 20 untiff = 1 timestamp char source char raray textMessage unsigned char messageType = ROBOT_MESSAGE = 20 untiff = 1 timestamp char source char raray textMessage unsigned char messageType = ROBOT_MESSAGE = 20 untiff = 1 timestamp char source char raray textMessage unsigned char messageType = ROBOT_MESSAGE = 20 untiff = 1 timestamp char source char raray textMessage unsigned char messageType = ROBOT_MESSAGE = 20 untiff = 1 timestamp char source char raray textMessage unsigned char messageType = ROBOT_MESSAGE = 20 untiff = 1 timestamp char source char raray textMessage unsigned char messageType = ROBOT_MESSAGE = 20 untiff = 1 timestamp char source char raray textMessage unsigned char messageType = ROBOT_MESSAGE = 20 untiff = 1 timestamp char source char raray textMessage unsigned char messageType = ROBOT_MESSAGE = 20 untiff = 1 timestamp char source char raray textMessage unsigned char messageType = ROBOT_MESSAGE = 20 untiff = 1 timestamp char source char raray textMessage unsigned char messageType = ROBOT_MESSAGE = 20 untiff = 1 timestamp char source char raray textMessage unsigned char messageType =	char source	<u> </u>			 			
int robotMessageArgument unsigned char titleSize char array ressageTitle char array ressageTitle char array textMessage char titleSize char array textMessage char array textMessage char messageSize char messageSize char source char robotMessageType = ROBOT_MESSAGE_PROGRAM_LABEL = 1 char array textMessage char messageSize char messageSize char messageSize char messageSize char array textMessage char messageType = ROBOT_MESSAGE_PROGRAM_LABEL = 1 char array textMessage char messageSize char messageSize char robotMessageType = ROBOT_MESSAGE_POPUP = 2 char robotMessageType = ROBOT_MESSAGE = 20 char robotMessage	char robotMessageType = ROBOT_ME	SSAGE_KEY	= 7					
unsigned char titleSize char array messageTitle char array textMessage LabelMessage int messageSize unsigned char messageType = ROBOT_MESSAGE = 20 uint64_t timestamp char source char robotMessage char robotMessage unsigned char messageSize unsigned char messageSize unsigned char messageType = ROBOT_MESSAGE PROGRAM_LABEL = 1 int it id char array textMessage unsigned char messageType = ROBOT_MESSAGE = 20 uint64_t timestamp char source char robotMessageType = ROBOT_MESSAGE = 20 uint64_t timestamp char source char robotMessageType = ROBOT_MESSAGE = 20 uint64_t timestamp char source char robotMessageType = ROBOT_MESSAGE = 20 uint64_t timestamp char source char robotMessageType = ROBOT_MESSAGE = 20 uint64_t timestamp char source char robotMessageType = ROBOT_MESSAGE = 20 uint64_t timestamp char source char robotMessageType = ROBOT_MESSAGE = 20 uint64_t timestamp char source char robotMessageType = ROBOT_MESSAGE = 20 uint64_t timestamp char source char robotMessageType = ROBOT_MESSAGE = 20 uint64_t timestamp char source char robotMessageType = ROBOT_MESSAGE = 20 uint64_t timestamp char source char robotMessageType = ROBOT_MESSAGE = 20 uint64_t timestamp char source char robotMessageType = ROBOT_MESSAGE = 20 uint64_t timestamp c	int robotMessageCode							
char array messageTitle char array textMessage LabelMessage Int messageSize Unsigned char messageType = ROBOT_MESSAGE = 20 Unitife4_t timestamp Int id Int id Int id Int id Int id Int id Int messageType = ROBOT_MESSAGE = 20 Unitife4_t timestamp Unitife4_t times	int robotMessageArgument							
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 unsigned char messageType = ROBOT_MESSAGE = 20 uint64_t timestamp char source char robotMessageType = ROBOT_MESSAGE = 20 uint64_t timestamp char source char robotMessageType = ROBOT_MESSAGE = 20 uint64_t timestamp char source char robotMessageType = ROBOT_MESSAGE_POPUP = 2 bool warning bool error unsigned char titleSize char array textMessage RequestValueMessage RequestValueMessage unsigned char messageType = ROBOT_MESSAGE = 20	unsigned char titleSize							
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 unsigned char messageType = ROBOT_MESSAGE = 20 uint64_t timestamp char source char robotMessageType = ROBOT_MESSAGE = 20 uint64_t timestamp char source char robotMessageType = ROBOT_MESSAGE = 20 uint64_t timestamp char source char robotMessageType = ROBOT_MESSAGE_POPUP = 2 bool warning bool error unsigned char titleSize char array textMessage RequestValueMessage RequestValueMessage unsigned char messageType = ROBOT_MESSAGE = 20	char array messageTitle							
int messageSize	char array textMessage							
int messageSize								
int messageSize	LabelMessage							
unsigned char messageType = ROBOT_MESSAGE = 20 uint64_t timestamp 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 = 20 uint64_t timestamp char source char robotMessageType = ROBOT_MESSAGE POPUP = 2 bool warning bool errori unsigned char titleSize char array messageTitle char array messageTitle char array textMessage RequestValueMessage unsigned char messageType = ROBOT_MESSAGE = 20 unsigned char titleSize char messageType = ROBOT_MESSAGE = 20 unsigned char titleSize char may textMessage unsigned char messageType = ROBOT_MESSAGE = 20	int messageSize							
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 ittleSize char array messageTitle char array messageTitle char array textMessage RequestValueMessage unsigned char messageType = ROBOT_MESSAGE = 20 ### Comparison of the Comparison of t	unsigned char messageType = ROBOT_	_MESSAGE =	: 20					
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 unsigned char messageType = ROBOT_MESSAGE = 20 unidea	uint64_t timestamp							
int id char array textMessage PopupMessage P	char source							
int id char array textMessage PopupMessage P	char robotMessageType = ROBOT_ME	SSAGE_PRO	GRAM_LAI	BEL = 1				
PopupMessage int messageSize unsigned char messageType = ROBOT_MESSAGE = 20 uint64_t timestamp char source char robotMessageType = ROBOT_MESSAGE_POPUP = 2 bool warning bool errori unsigned char titleSize char array messageTitle char array textMessage RequestValueMessage int messageSize unsigned char messageType = ROBOT_MESSAGE = 20	int id							
PopupMessage int messageSize unsigned char messageType = ROBOT_MESSAGE = 20 uint64_t timestamp char source char robotMessageType = ROBOT_MESSAGE_POPUP = 2 bool warning bool errori unsigned char titleSize char array messageTitle char array textMessage RequestValueMessage int messageSize unsigned char messageType = ROBOT_MESSAGE = 20	char array textMessage							
int messageSize unsigned char messageType = ROBOT_MESSAGE = 20 uint64_t timestamp char source char robotMessageType = ROBOT_MESSAGE_POPUP = 2 bool warning bool errori unsigned char titleSize char array messageTitle char array textMessage int messageSize unsigned char messageType = ROBOT_MESSAGE_POPUP = 2								
int messageSize unsigned char messageType = ROBOT_MESSAGE = 20 uint64_t timestamp char source char robotMessageType = ROBOT_MESSAGE_POPUP = 2 bool warning bool errori unsigned char titleSize char array messageTitle char array textMessage int messageSize unsigned char messageType = ROBOT_MESSAGE_POPUP = 2	PopupMessage							
uint64_t timestamp char source char robotMessageType = ROBOT_MESSAGE_POPUP = 2 bool warning bool error unsigned char titleSize char array messageTitle char array textMessage int messageSize unsigned char messageType = ROBOT_MESSAGE = 20	int messageSize							
char source char robotMessageType = ROBOT_MESSAGE_POPUP = 2 bool warning bool error unsigned char titleSize char array messageTitle char array textMessage lint messageSize unsigned char messageType = ROBOT_MESSAGE = 20	unsigned char messageType = ROBOT_	_MESSAGE =	: 20					
char robotMessageType = ROBOT_MESSAGE_POPUP = 2 bool warning bool error unsigned char titleSize char array messageTitle char array textMessage char array textMessage RequestValueMessage int messageSize unsigned char messageType = ROBOT_MESSAGE = 20	uint64_t timestamp							
bool error unsigned char titleSize char array messageTitle char array textMessage RequestValueMessage int messageSize unsigned char messageType = ROBOT_MESSAGE = 20	char source							
bool error unsigned char titleSize char array messageTitle char array textMessage RequestValueMessage int messageSize unsigned char messageType = ROBOT_MESSAGE = 20	char robotMessageType = ROBOT_ME	SSAGE_POP	UP = 2					
bool error unsigned char titleSize char array messageTitle char array textMessage RequestValueMessage int messageSize unsigned char messageType = ROBOT_MESSAGE = 20	bool warning							
char array messageTitle char array textMessage Char array textMessage RequestValueMessage int messageSize unsigned char messageType = ROBOT_MESSAGE = 20	bool error							
Char array textMessage RequestValueMessage int messageSize unsigned char messageType = ROBOT_MESSAGE = 20	unsigned char titleSize							
RequestValueMessage int messageSize unsigned char messageType = ROBOT_MESSAGE = 20	char array messageTitle							
int messageSize unsigned char messageType = ROBOT_MESSAGE = 20	char array textMessage							
int messageSize unsigned char messageType = ROBOT_MESSAGE = 20								
int messageSize unsigned char messageType = ROBOT_MESSAGE = 20	RequestValueMessage							
unsigned char messageType = ROBOT_MESSAGE = 20 unt64_t timestamp	int messageSize							
uint64_t timestamp	unsigned char messageType = ROBOT_	_MESSAGE =	: 20				Y	
	uint64_t timestamp							

char robotMessageType = ROBOT_MESSAGE_REQUEST unsigned int requestId char array textMessage TextMessage int messageSize unsigned char messageType = ROBOT_MESSAGE = 20 uint64_t timestamp					
unsigned int requestId char array textMessage TextMessage int messageSize unsigned char messageType = ROBOT_MESSAGE = 20					
char array textMessage TextMessage int messageSize unsigned char messageType = ROBOT_MESSAGE = 20					
TextMessage int messageSize unsigned char messageType = ROBOT_MESSAGE = 20					
int messageSize unsigned char messageType = ROBOT_MESSAGE = 20					
unsigned char messageType = ROBOT_MESSAGE = 20					
unsigned char messageType = ROBOT_MESSAGE = 20					
				-	
		:			
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_VARIABLI	.E = 8				
int code					
int argument					
unsigned char titleSize					
char array messageTitle					
char array messageText					
Example of data recieved: Example of data recieved:	ample				

3.0 Retur til ind	ex							
Primary and s	econdary	client						
The robot state messa	ge contain sever	al nackages	as soon he	low				
The robot state messa	ge contain sever	i packages	as seen be	iow.				
NOTE:								
 Not every robot state 								
			pe chars, t	o see what pa	ckages are included in the mess	age.		
 Not all package type 								
Only the message ty	oe "ROBOT_STA	ΓΕ" is docun	nented her	e (messageTy	pe value is 16)			
							<u>.</u>	
B				ata, cartesian	info, kinematics info, masterboa	ird data, tool dat	a, configura	tion data)
unsigned char message	eType = ROBOT_	STATE = 16						
This page describes the	e data sent from	the robot c	ontroller:		DataStreamFromURController			
Pohot modo o	lata							
Robot mode o	lata							
int packageSize	<u>.</u>		<u>.</u>					
unsigned char package uint64 t timestamp	: Type = ROBO1_	VIODE_DAT	A = U					
bool isRobotConnecte	١							
bool isRealRobotEnabl								
bool isPowerOnRobot	eu							
bool isEmergencyStop	ned							
bool isProtectiveStopp								
bool isProgramRunnin	•••••						<u> </u>	
bool isProgramPaused	<u>-</u>							
unsigned char robotM	ode		•					
unsigned char control								
double targetSpeedFra	ıction							
double speedScaling								

Joint data							¿	
int packageSize							(
unsigned char packageType =	JOINT_DA	TA = 1						
for each joint:								
double q_actual								
double q_target								
double qd_actual float I_actual								
float I_actual								
float V_actual								
float T_motor						<u> </u>	<u>.</u>	
float T_micro								
unsigned char jointMode							<u>.</u>	
end							<u>.</u>	
The "jointMode" field is a co	de for the j	oint status	(shown on	the initialisat	ion screen):	<u>JointModes</u>		
							į	
Cartesian info								
int packageSize								
unsigned char = CARTESIAN_	INFO = 4							
double X							<u></u>	
double Y								
double Y double Z							<u> </u>	
double Rx						<u> </u>		
double Ry								
double Rz								,
Masterboard data								
int packageSize								
unsigned char packageType =	= MASTERB	OARD_DA	ΓA = 3					
int digitalInputBits								
int digitalOutputBits								
char analogInputRange0								

re e							
e							
alled							
alled, also	the follow	ing:					
obots soft	ware only)						
		••••••					
TOOL_DA	TA = 2						
		••••••					
age							
	e Illed alled, also boots soft	e Illed alled, also the follow bobots software only)	e Illed alled, also the following: bbots software only) TOOL_DATA = 2	e Illed alled, also the following: Dibots software only) TOOL_DATA = 2	e illed alled, also the following: bbots software only) TOOL_DATA = 2	e alled alled, also the following: Dibots software only) TOOL_DATA = 2	e illed alled, also the following: Dibots software only): TOOL_DATA = 2

			-	<u> </u>	:	•
Kinematics info						
int packageSize						
unsigned char packageType = KINEI	MATICS INFO = 5					
i i						
This package contains a checksum f	or the specific rob	ot. It might be subi	iect of change in the near future.			
					ļ	
Configuration data						
int packageSize						
unsigned char packageType = CONF	i	\ = 6				
for each joint:						
double jointMinLimit					<u> </u>	
double jointMaxLimitt					<u> </u>	
for each joint:	i i					
double jointMaxSpeed						
double jointMaxAcceleration						}
double vJointDefault						
double a Joint Default						
double vToolDefault						
double aToolDefault						
double eqRadius						
for each joint:						
double DHa						
for each joint:						
double DHd						
for each joint:						,
double DHalpha					<u>.</u>	
for each joint:					<u> </u>	
double DHtheta						
int masterboardVersion						
int controllerBoxType						
int robotType						
int robotSubType						

	1 1	
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 sub	ect 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		
The source ficture actual of the message.		
VocionMoccago first packago only		
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			
		į		
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								
, , , , , , , , , , , , , , , , , , , ,								
The "warningLevel" field can	contain the	 following	values:		WarningLevels			
KeyMessage								
int messageSize								
unsigned char messageType =	= ROBOT_N	лessage =	20					
uint64_t timestamp								
char source								
char robotMessageType = RO	BOT_MESS	SAGE_KEY	= 7					
int robotMessageCode								
int robotMessageArgument								
unsigned char titleSize								
char array messageTitle								
char array textMessage							<u></u>	
							: : :	
LabelMessage								
int messageSize								
unsigned char messageType =	= ROBOT_N	/IESSAGE =	20					
uint64_t timestamp								
char source								
char robotMessageType = RO	BOT_MESS	SAGE_PRO	GRAM_LAE	BEL = 1				
int id								
char array textMessage								
<u>į</u>								
RequestValueMess	sage							
int messageSize								
unsigned char messageType =	= ROBOT_N	/IESSAGE =	20					
uint64_t timestamp								
char source								
char robotMessageType = RO	BOT_MESS	SAGE_REQI	UEST_VALU	JE = 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 thi	s package,	if the 'requ	estedType'	field contains	s the value 8. This is the 'PopupMe	ssage' type.		
TextMessage								
int messageSize								
unsigned char messageType	= ROBOT_N	MESSAGE =	20					
uint64_t timestamp								
char source								
char robotMessageType = RC	DBOT_MESS	SAGE_TEXT	= 0					
char array textMessage								
]							<u>.</u>	
RuntimeException	าMessa	ge						
int messageSize								
unsigned char messageType	= ROBOT_N	MESSAGE =	20					
uint64_t timestamp								
char source								
char robotMessageType = RC	DBOT_MESS	SAGE_RUN	TIME_EXCE	PTION = 10				
char array textMessage								
VarMessage								
int messageSize								
unsigned char messageType	= PROGRAN	M_STATE_I	MESSAGE =	25				
uint64_t timestamp								
char robotMessageType = PR	OGRAM_S	TATE_MES	SAGE_VARI	ABLE_UPDAT	E = 2			
unsigned char titleSize								
char array messageTitle								
char array messageText								

Deprecated:Use the much more efficient 'GlobalVariablesSetupMessage' and 'GlobalVariablesUpdateMessa	ge' (described be	elow).	? : :
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 "	/alueType"	field are:		<u>ValueTypes</u>			
ļ <u>.</u>							
The information for each var	iable is on	a new line	(terminate	d with the '\n'	character).		
Example of data reciev	ed:		Example	<u></u>			

2 4						İ	:		<u> </u>
3.1	Retur til index								
Primar	y and seco	ndary	client						
The robot	state message cor	ntain sever	al packages	s as seen be	elow.				
NOTE:									
Not ever	y robot state mes	sage may n	necessarily	contain all	of the packages	s described.			
				ype chars,	to see what pa	ckages are included in the message			
	ackage types are o							<u>.</u>	
 Only the 	message type "Ro	OBOT_STAT	ΓΕ" is docu	mented he	re (messageTyp	pe value is 16)			
								<u></u>	
					lata, cartesian i	nfo, kinematics info, masterboard o	lata, tool data,	configurati	on data)
unsigned c	har messageType	= ROBOT_	STATE = 16						
This page of	describes the data	sent from	the robot	controller:		<u>DataStreamFromURController</u>		<u> </u>	
						<u> </u>		<u> </u>	
	mode data								
int package									
	har packageType	= ROBOT_[MODE_DAT	ΓA = 0				<u></u>	
uint64_t ti									
·	otConnected							<u> </u>	
	lRobotEnabled								
	verOnRobot								
	ergencyStopped								
	tectiveStopped					<u> </u>		<u> </u>	
	gramRunning							<u>.</u>	
	gramPaused								
	char robotMode							<u> </u>	
	char controlMode								
	getSpeedFraction								
double spe	eedScaling								

	<u> </u>	· · · · · · · · · · · · · · · · · · ·			
Joint data					
int packageSize					
unsigned char packageType =	JOINT_DATA = 1				
for each joint:					
double q_actual					
double q_target					
double q_actual double q_target double qd_actual					
float I_actual					
float V_actual					
float T_motor					
float T_motor float T_micro unsigned char jointMode					
unsigned char jointMode					
end					
The "jointMode" field is a cod	e for the joint status (shov	n on the initialisation scre	en):	<u>Joint Modes</u>	
Cartesian info					
int packageSize					
unsigned char = CARTESIAN_II	NFO = 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	3					
int packageSize						
unsigned char packageType	= MASTER	BOARD_DA	TA = 3			
int digitalInputBits						
int digitalOutputBits						
char analogInputRange0						
char analogInputRange1						
double analogInput0						
double analogInput1						
char analogOutputDomain0	J					
char analogOutputDomain1						
double analogOutput0						
double analogOutput1						
float masterBoardTemperat	.ure					
float robotVoltage48V						
float robotCurrent						
float masterIOCurrent						
unsigned char safetyMode						
unsigned char InReducedMo	ode					
char euromap67InterfaceIns	stalled					
(if euromap67 interface is in	ıstalled, als	o the follov	wing:			
int euromapInputBits						
int euromapOutputBits						
float euromapVoltage						
float euromapCurrent)						
uint32_t (Used by Universal	Robots sof	tware only)			
Tool data						
int packageSize						
unsigned char packageType	= TOOL_D/	ATA = 2				
char analogInputRange2						
char analogInputRange3						
double analogInput2						

double analogInput3							
float toolVoltage48V							
unsigned char toolOutputVo	oltage						
float toolCurrent							
float toolTemperature							
unsigned char toolMode							
Kinematics info							
int packageSize							
unsigned char packageType	= KINEMAT	ICS_INFO =	= 5				
This package contains a che	cksum for t	he specific	robot. It m	ight be subject	of change in the near future.		
Configuration dat	ta						
int packageSize							
unsigned char packageType	= CONFIGU	IRATION_D	ATA = 6				
for each joint:							
double jointMinLimit							
double jointMaxLimitt							
for each joint:							
double jointMaxSpeed							
double jointMaxAcceleration	n						
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_MO	DE_DATA =	= 7						
double X								
double Y								
double Z								
double Rx								
double Ry								
double Rz								
double robotDexterity								
Additional info								
int packageSize								
unsigned char = ADDITIONA	$L_{INFO} = 8$							
bool teachButtonPressed								
bool teachButtonEnabled								
Calibration data								
int packageSize								
unsigned char packageType	= CALIBRA	TION_DATA	\ = 9					
This package is used interna	lly by Unive	ersal Robot	s software	only and should	d be skipped. It might be subject of	change in the r	ear future.	
VesionMessage -	first pa	ackage	only					
	•		,					
						:	•	

This is the first package sent on both t	he primary and seconda	ry 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_ME	SSAGE_VERSION = 3			
char projectNameSize				
char array projectName				
unsigned char majorVersion				
unsigned char minorVersion				
int svnRevision				
char array buildDate				
Below is only for prima	ry client			
The "source" field is a code for the sei	nder of the message:	MessageSources		
Zero or more of the following messag	es:			
SafetyModeMessage				
int messageSize				
unsigned char messageType = ROBOT	MESSAGE = 20			
uint64_t timestamp				
char source				
char robotMessageType = ROBOT_ME	SSAGE_SAFETY_MODE	5		
int robotMessageCode				
int robotMessageArgument				
char safetyModeType				
char array textMessage				
The "safetyModeType" field can conta	in the following values:	SafetyModeTypes		
RobotcommMessage				

int messageSize						
unsigned char messageType = ROBOT	_MESSAGE =	20				
uint64_t timestamp						
char source						
char robotMessageType = ROBOT_MI	ESSAGE_ERRC	R_CODE = 6				
int robotMessageCode						
int robotMessageArgument						
int warningLevel						
char array textMessage		į				
The "warningLevel" field can contain	the following	values:		WarningLevels		
		Ī				
KeyMessage		i				
int messageSize						
unsigned char messageType = ROBOT	MESSAGE =	20				
uint64_t timestamp						
char source						
char robotMessageType = ROBOT_MI	ESSAGE KEY =	= 7				
int robotMessageCode			•••••			
int robotMessageArgument						
unsigned char titleSize		i				
char array messageTitle						
char array textMessage						
LabelMessage						
int messageSize						
unsigned char messageType = ROBOT	MESSAGE =	20				
uint64_t timestamp						
char source	i	·····				
char robotMessageType = ROBOT_MI	ESSAGE PROC	GRAM LABEL	= 1			
int id		-				
char array textMessage						
,	<u> </u>	·····				<u></u>
<u> </u>		i			 <u> </u>	<u> </u>

RequestValueMessage					
int messageSize					
unsigned char messageType = ROBOT_MI	ESSAGE = 20				
uint64_t timestamp					
char source					
char robotMessageType = ROBOT_MESSA	AGE_REQUEST_VALU	JE = 9			
unsigned int requestId					
unsigned int requestedType = 0-8	<u> </u>				
(*) bool warning					
(*) bool error					
(*) bool blocking					
(*) unsigned char titleLength					
(*) char array messageTitle					
char array textMessage					
	<u> </u>				
(*) Fields only included in this package, if	the 'requestedType	' field contains	the value 8. This is the 'PopupMessa	age' type.	
TextMessage					
int messageSize					
unsigned char messageType = ROBOT_MI	ESSAGE = 20				
uint64_t timestamp	<u> </u>				
char source					
char robotMessageType = ROBOT_MESSA	AGE_TEXT = 0				
char array textMessage					
RuntimeExceptionMessag	je 💮				
int messageSize					
unsigned char messageType = ROBOT_MI	ESSAGE = 20				
uint64_t timestamp					
char source					
char robotMessageType = ROBOT_MESSA	AGE_RUNTIME_EXC	EPTION = 10			
char array textMessage					

int messageSize unsigned char messageType = PROGRAM_STATE_MESSAGE = 25 unsigned char itileSize char array messageType = PROGRAM_STATE_MESSAGE_VARIABLE_UPDATE = 2 unsigned char itileSize 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 = 25 uint64_t timestamp char robotMessageType = 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 short startIndex for each variable: (terminated with the *\n' character) 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									
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 messageType = PROGRAM_STATE_MESSAGE_data_variablesSetupMessage' 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 (*) Char array variableNames (*) Each new line (terminated with the '\n' character) in the array contains the name of a variable. GlobalVariablesUpdateMessage 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 short startIndex for each variable: (terminated with the '\n' character) 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	VarMessage								
uint64_t timestamp	int messageSize								
uint64_t timestamp	unsigned char messageType	= PROGRAI	M_STATE_I	MESSAGE =	= 25				
unsigned char titleSize char array messageTitle 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 (*) thar 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 unit64_t timestamp char robotMessageType = PROGRAM_STATE_MESSAGE = 25 unit64_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	uint64_t timestamp								
unsigned char titleSize char array messageTitle 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 (*) thar 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 unit64_t timestamp char robotMessageType = PROGRAM_STATE_MESSAGE = 25 unit64_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	char robotMessageType = PF	ROGRAM_S	TATE_MES	SAGE_VAR	RIABLE_UPDATE	= 2			
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 unsigned char messageType = PROGRAM_STATE_MESSAGE = 25 unsigned short startIndex for each variable: (terminated with the '\n' character) 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	unsigned char titleSize								
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 (*) char array variableNames GlobalVariablesUpdateMessage int messageSize unsigned char messageType = PROGRAM_STATE_MESSAGE_SETUP = 0 unsigned char messageType = PROGRAM_STATE_MESSAGE = 25 unit64_t timestamp char robotMessageType = PROGRAM_STATE_MESSAGE = 25 unit64_t timestamp for each variable: (terminated with the '\n' character) 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	char array messageTitle								
Global Variables Setup Message int message Size unsigned char message Type = PROGRAM_STATE_MESSAGE = 25 unitn64_t timestamp char robot Message Type = 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. Global Variable SUpdate Message int message Size unsigned char message Type = PROGRAM_STATE_MESSAGE = 25 unit64_t timestamp char robot Message Type = PROGRAM_STATE_MESSAGE = Clobal_Variable (terminated with the '\n' character) unsigned short startIndex for each variable: (terminated with the '\n' character) unsigned char value Type One of following data chunks based on the value of 'value Type': value Type = 'NONE_VAL': // No content -> has not been initialized yet	char array messageText	<u> </u>							
Global Variables Setup Message int message Size unsigned char message Type = PROGRAM_STATE_MESSAGE = 25 unitn64_t timestamp char robot Message Type = 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. Global Variable SUpdate Message int message Size unsigned char message Type = PROGRAM_STATE_MESSAGE = 25 unit64_t timestamp char robot Message Type = PROGRAM_STATE_MESSAGE = Clobal_Variable (terminated with the '\n' character) unsigned short startIndex for each variable: (terminated with the '\n' character) unsigned char value Type One of following data chunks based on the value of 'value Type': value Type = 'NONE_VAL': // No content -> has not been initialized yet							<u> </u>		
Global Variables Setup Message int message Size unsigned char message Type = PROGRAM_STATE_MESSAGE = 25 unitn64_t timestamp char robot Message Type = 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. Global Variable SUpdate Message int message Size unsigned char message Type = PROGRAM_STATE_MESSAGE = 25 unit64_t timestamp char robot Message Type = PROGRAM_STATE_MESSAGE = Clobal_Variable (terminated with the '\n' character) unsigned short startIndex for each variable: (terminated with the '\n' character) unsigned char value Type One of following data chunks based on the value of 'value Type': value Type = 'NONE_VAL': // No content -> has not been initialized yet	Deprecated:Use the much m	ore efficier	nt 'GlobalV	ariablesSet	upMessage' ar	nd 'GlobalVariablesUpdateMessage'	(described belo	ow).	
int messageSize unsigned char messageType = PROGRAM_STATE_MESSAGE = 25 uint64_t timestamp char robottMessageType = 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 robottMessageType = 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		į							
int messageSize unsigned char messageType = PROGRAM_STATE_MESSAGE = 25 uint64_t timestamp char robottMessageType = 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 robottMessageType = 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	GlobalVariablesSe	etupMe	ssage						
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	int messageSize								
char robotMessageType =PROGRAM_STATE_MESSAGE_GLOBAL_VARIABLES_SETUP = 0 unsigned short startIndex	unsigned char messageType	= PROGRAI	M_STATE_I	MESSAGE =	= 25				
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	uint64_t timestamp								
(*) Char array variableNames (*) Each new line (terminated with the '\n' character) in the array contains the name of a variable. GlobalVariablesUpdateMessage int messageSize unusigned 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	char robotMessageType =PR	OGRAM_ST	TATE_MESS	SAGE_GLO	BAL_VARIABLE	S_SETUP = 0			
(*) 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 startlindex 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	unsigned short startIndex								
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	(*) char array variableNames	3							
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									
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	(*) Each new line (terminate	d with the '	'\n' charact	ter) in the a	array contains t	he name of a variable.			
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		<u></u>							
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	GlobalVariablesUp	pdateM	essage	3					
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	int messageSize								
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	unsigned char messageType	= PROGRAI	M_STATE_I	MESSAGE =	= 25				
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	uint6/Lt timestamn								
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	char robotMessageType = PF	ROGRAM_S	TATE_MES	SAGE_GLO	BAL_VARIABLE	S_UPDATE = 1			
unsigned char valueType One of following data chunks based on the value of 'valueType': valueType = 'NONE_VAL': // No content -> has not been initialized yet	unsigned short startIndex								
unsigned char valueType One of following data chunks based on the value of 'valueType': valueType = 'NONE_VAL': // No content -> has not been initialized yet	for each variable: (terminate	d with the	'\n' charac	ter)					
valueType = 'NONE_VAL': // No content -> has not been initialized yet	unsigned char valueType								
// No content -> has not been initialized yet		s based on	the value c	of 'valueTyp	oe':				
	valueType = 'NONE_VAL':								
valueType = 'STRING_VAL':	// No content -> has not bee	n initialized	l 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:	<u>ValueTypes</u>			
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								
l	y and seco	ndarv	client						
	, and sees	ii.aai y	0110110						
The robot	state message coi	ntain sever	al packages	as seen be	:elow.				
									
NOTE:									
• Not ever	y robot state mes	sage may n	ecessarily	contain all	of the packages	described.			
Therefor	e, it is a good idea	a to test the	e package t	ype chars,	to see what pad	ckages are included in the messag	e.		
 Not all p 	ackage types are o	documente	d here.						
 Only the 	message type "Ro	OBOT_STAT	TE" is docur	mented he	re (messageTyp	e value is 16)			
					lata, cartesian i	nfo, kinematics info, masterboard	data, tool data,	configurati	on data)
unsigned o	char messageType	= ROBOT_	STATE = 16						
									
This page	describes the data	sent from	the robot (controller:		DataStreamFromURController			
Robot	mode data								
int packag	************************************								
	char packageType	= ROBOT_[MODE_DAT	A = 0					
uint64_t t	.								
	otConnected								
	lRobotEnabled								
	verOnRobot								
	ergencyStopped								
	tectiveStopped								
	gramRunning								
	gramPaused								
	char robotMode								
·	char controlMode	:							
	getSpeedFraction								
double spe	eedScaling								

double targetSpeedFractionL	imit						
						¢	}
Joint data							
int packageSize							
unsigned char packageType =	= JOINT_DATA = 1						
for each joint: double q_actual double q_target							
double q_actual						ç	
double q_target							
double qd_actual							
float I_actual							
float V_actual							
float T_motor							
float T_motor float T_micro							
unsigned char jointMode							
end							
			<u> </u>				
The "jointMode" field is a co	de for the joint st	itus (shown o	n the initialisation	on screen):	<u>JointModes</u>		
Cartesian info							
int packageSize						<u> </u>	
unsigned char = CARTESIAN_	INFO = 4						
double X							
double Y							
double Z							
double Rx						<u> </u>	
double Ry							
double Rz			<u> </u>				
double TCPOffsetX			<u> </u>			<u> </u>	
double TCPOffsetY						<u> </u>	
double TCPOffsetZ							
double TCPOffsetRx						ļ	
double TCPOffsetRy			<u> </u>			<u> </u>	
double TCPOffsetRz							

Mactorboard data			i 	
Masterboard data				
int packageSize				
unsigned char packageType = MASTERBOARD_D	ATA = 3	<u>:</u>		
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 follo	owing:			
int euromapInputBits				
int euromapOutputBits				
float euromapVoltage				
float euromapCurrent)				
uint32_t (Used by Universal Robots software onl	y)			
uint8_t operational Mode Selector Input				
uint8_t threePositionEnablingDeviceInput				
Tool data				
int packageSize				
unsigned char packageType = TOOL_DATA = 2				

1 1 1 15 5	:			:		•	
char analogInputRange2					į	 :	į
char analogInputRange3							
double analogInput2							
double analogInput3							
float toolVoltage48V							
unsigned char toolOutputV	oltage						
float toolCurrent						:	
float toolTemperature						<u> </u>	
unsigned char toolMode							
						<u> </u>	
Kinematics info							
int packageSize							
unsigned char packageType	e = KINEMAT	ΓICS_INFO	= 5				
•	1 :		•				
This package contains a che	cksum for t	he specific	robot. It m	night be subject	of change in the near future.		
This information is sent wh	en leaving i	nitializing n	node and/o	or if the kinema	tics configuration is changed.		
		nitializing n	node and/o	or if the kinema	of change in the near future. tics configuration is changed.		
		nitializing n	node and/o	or if the kinema	tics configuration is changed.		
Configuration da		nitializing n	node and/o	or if the kinema	tics configuration is changed.		
Configuration da int packageSize	ta			or if the kinema	tics configuration is changed.		
Configuration da int packageSize unsigned char packageType	ta			or if the kinema	tics configuration is changed.		
Configuration da int packageSize unsigned char packageType for each joint:	ta			or if the kinema	tics configuration is changed.		
Configuration da int packageSize unsigned char packageType	ta			or if the kinema	tics configuration is changed.		
Configuration da int packageSize unsigned char packageType for each joint: double jointMinLimit	ta			or if the kinema	tics configuration is changed.		
Configuration da int packageSize unsigned char packageType for each joint: double jointMinLimit double jointMaxLimitt for each joint:	ta			or if the kinema	tics configuration is changed.		
Configuration da int packageSize unsigned char packageType for each joint: double jointMinLimit double jointMaxLimitt for each joint: double jointMaxSpeed	ta = CONFIGU			or if the kinema	tics configuration is changed.		
Configuration da int packageSize unsigned char packageType for each joint: double jointMinLimit double jointMaxLimitt for each joint:	ta = CONFIGU			or if the kinema	tics configuration is changed.		
Configuration da int packageSize unsigned char packageType for each joint: double jointMinLimit double jointMaxLimitt for each joint: double jointMaxSpeed double jointMaxAcceleratio	ta = CONFIGU			or if the kinema	tics configuration is changed.		
Configuration da int packageSize unsigned char packageType for each joint: double jointMinLimit double jointMaxLimitt for each joint: double jointMaxSpeed double jointMaxAcceleratio	ta = CONFIGU			or if the kinema	tics configuration is changed.		
Configuration da int packageSize unsigned char packageType for each joint: double jointMinLimit double jointMaxLimitt for each joint: double jointMaxSpeed double jointMaxAcceleration double vJointDefault double aJointDefault	ta = CONFIGU			or if the kinema	tics configuration is changed.		
Configuration da int packageSize unsigned char packageType for each joint: double jointMinLimit double jointMaxLimitt for each joint: double jointMaxSpeed double jointMaxAcceleration double vJointDefault double aJointDefault double vToolDefault	ta = CONFIGU			or if the kinema	tics configuration is changed.		
Configuration da int packageSize unsigned char packageType for each joint: double jointMinLimit double jointMaxLimitt for each joint: double jointMaxSpeed double jointMaxAcceleration double vJointDefault double aJointDefault double aToolDefault	ta = CONFIGU			or if the kinema	tics configuration is changed.		

double DHd for each joint: double DHalpha for each joint: double DHtheta int masterboardVersion int controllerBoxType int robotType int robotSubType int robo						
for each joint: double DHalpha for each joint: double DHtheta int masterboardVersion int controllerBoxType int robotType int robotType int robotSubType int robotSubType Int in masterboard in it 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 X unused double Y unused double Rx unused double Rx unused double Rx unused double Rx unused double Rq	for each joint:					
double DHalpha for each joint: double DHtheta Int masterboardVersion Int controllerBoxType Int robotSubType	double DHd					
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 X unused double Y unused double Rx unused double Rx unused double Rx unused double Rx unused double Rz double Rz double robotDexterity	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 Y unused double Rx unused double Rx unused double Rx unused double Ry unused double Rz double robotDexterity	double DHalpha					
int masterboardVersion int controllerBoxType int robotType int robotSubType 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 Y unused double Rx unused double Rx unused double Rx unused double Rz unused double Rz double robotDexterity	for each joint:					
int controllerBoxType int robotSubType 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 Y unused double Rx unused double Rx unused double Rx unused double Rx unused double Ry unused double Rz double robotDexterity	double DHtheta					
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 Y unused double Rx unused double Rx unused double Rx unused double Ry unused double Rz double robotDexterity	int masterboardVersion					
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 Y unused double Rx unused double Rx unused double Rx unused double Ry unused double Rz double robotDexterity	int controllerBoxType					
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 Y unused double Rx unused double Rx unused double Rx unused double Ry unused double Ry unused double Ry unused double Rz double robotDexterity						
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 Rx unused double Ry unused double Ry unused double Rz double robotDexterity	int robotSubType					
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 Rx unused double Ry unused double Ry unused double Rz double robotDexterity					<u> </u>	
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 Rx unused double Ry unused double Ry unused double Rz double robotDexterity	This information is sent when leaving ir	nitializing mode ar	nd/or if the kinema	tics configuration is changed.		
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 Ry unused double Rz double robotDexterity						
unused double X unused double Y unused double Z unused double Rx unused double Ry unused double Ry unused double Rz double robotDexterity	Force mode data					
unused double X unused double Y unused double Z unused double Rx unused double Ry unused double Ry unused double Rz double robotDexterity	int packageSize					
unused double X unused double Y unused double Z unused double Rx unused double Ry unused double Ry unused double Rz double robotDexterity	unsigned char = FORCE_MODE_DATA =	· 7				
unused double Z unused double Rx unused double Ry unused double Ry unused double Rz double robotDexterity	unused double X					
unused double Rx unused double Ry unused double Rz double robotDexterity	unused double Y					
unused double Ry unused double Rz double robotDexterity	unused double Z					
double robotDexterity double robotDexterity						
double robotDexterity double robotDexterity	unused double Ry					
	unused double Rz					
	double robotDexterity					
					<u> </u>	
Additional info	Additional info					
int packageSize						
unsigned char = ADDITIONAL_INFO = 8	unsigned char = ADDITIONAL_INFO = 8					
bool freedriveButtonPressed	bool freedriveButtonPressed					
bool freedriveButtonEnabled	bool freedriveButtonEnabled				:	
bool IOEnabledFreedrive	bool IOEnabledFreedrive					
Calibration data	Calibration data					
int packageSize						

unsigned char packageType = CAL	IBRATION_DATA	= 9				
This package is used internally by	Universal Robots	software only and	should be skipped. It might	be subject of change in the	e near future	•
VersionMessage - firs	st package	only				
This is the first package sent on bo	oth the primary a	nd secondary clien	t interfaces. This package it	not part of the robot state	message.	
int messageSize						
unsigned char messageType = ROE	BOT_MESSAGE =	20				
uint64_t timestamp						
char source						
char robotMessageType = ROBOT_	_MESSAGE_VERS	ION = 3				
char projectNameSize						
char array projectName		<u> </u>				
unsigned char majorVersion		<u></u>				
unsigned char minorVersion						
int svnRevision		<u></u>				
char array buildDate		<u> </u>				
<u></u>						
Below is only for prin						
The "source" field is a code for the	sender of the m	essage:	<u>MessageSources</u>			
Zero or more of the following mes	sages:	<u> </u>				
SafetyModeMessage						
int messageSize						
unsigned char messageType = ROE	BOT_MESSAGE =	20				
uint64_t timestamp		i				
char source		į				
char robotMessageType = ROBOT_	_MESSAGE_SAFE	TY_MODE = 5				
int robotMessageCode						
int robotMessageArgument						

char safety Mode Type			
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:	<u>WarningLevels</u>		
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			

		:					:	1	: 1
uint64_t timestamp									
char source									
char robotMessageType = R	OBOT_MES	SAGE_PRO	GRAM_LA	BEL = 1					
int id									
char array textMessage								<u>.</u>	
RequestValueMes	ssage								
int messageSize									
unsigned char messageType	= ROBOT_	MESSAGE =	= 20						
uint64_t timestamp									
char source									
char robotMessageType = R	OBOT_MES	SAGE_REQ	UEST_VAL	UE = 9					
unsigned int requestId									
unsigned int requestedType	= 0-8								
(*) bool warning									
(*) bool error									
(*) bool blocking									
(*) unsigned char titleLength	1								
(*) char array messageTitle									
char array textMessage									
(*) Fields only included in th	is package,	if the 'reqા	uestedType	e' field contains	the value 8. This is th	e 'PopupMess	age' type.		
TextMessage									
int messageSize									
unsigned char messageType	= ROBOT_	MESSAGE =	= 20						
uint64_t timestamp									
char source									
char robotMessageType = R	OBOT_MES	SAGE_TEX	T = 0						
char array textMessage									
RuntimeException	nMessa	ige							
int messageSize		-							
	•	•					-	-	-

unsigned char messageType	e = ROBOT_	MESSAGE =	= 20						
uint64_t timestamp									
char source									
char robotMessageType = R	OBOT_MES	SAGE_RUN	ITIME_EXC	EPTION = 10				<u> </u>	
char array textMessage									
								<u> </u>	
VarMessage									
int messageSize									
unsigned char messageType	e = PROGRA	M_STATE_	MESSAGE :	= 25					
uint64_t timestamp									
char robotMessageType = P	ROGRAM_S	STATE_MES	SAGE_VAF	NABLE_UPDATE	= 2				
unsigned char titleSize									
char array messageTitle									
char array messageText									
Deprecated:Use the much n	nore efficie	nt 'GlobalV	ariablesSet	upMessage' ar	d 'GlobalVariablesU	lpdateMessage'	(described belo	ow).	
									
GlobalVariablesS	etupMe	essage							
int messageSize									
unsigned char messageType	e = PROGRA	M_STATE_	MESSAGE :	= 25					
uint64_t timestamp									
char robotMessageType =PI	ROGRAM_S	TATE_MES	SAGE_GLO	BAL_VARIABLE	S_SETUP = 0				
unsigned short startIndex									
(*) char array variableName	es							<u> </u>	
(*) Each new line (terminate	ed with the	'\n' charac	ter) in the a	array contains t	he name of a variab	le.		<u>.</u>	
GlobalVariablesU	pdateM	lessage	9						
int messageSize									
unsigned char messageType	e = PROGRA	M_STATE_	MESSAGE :	= 25					
uint64_t timestamp									
char robotMessageType = P	ROGRAM_S	STATE_MES	SAGE_GLC	BAL_VARIABLE	S_UPDATE = 1				
unsigned short startIndex									

for each variable: (terminated with the '\n' charac	ter)			
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:	<u>ValueTypes</u>			
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									
Primar	y and second	dary clier	nt							
The follow	ng messages have be	een changed w	ith respect	to the prev	ious major s	oftware rel	ease 3.2			
Please con	sult SW3.2 documen	tation for all o	ther packag	es.						
								,		
				<u> </u>						
Vesion	Message - fir	rst packa	ge only	/						
This is the	first package sent on	both the prim	ary and sec	ondary clie	nt interfaces	s. This packa	ige it not pa	rt of the ro	bot state m	essage.
int messag										
.	har messageType = R	ROBOT_MESSA	GE = 20							
uint64_t ti	mestamp			, ,		,				,
char source	-									
	MessageType = ROB(OT_MESSAGE_	VERSION =	3						
	tNameSize									
	projectName	,		,		,				,
unsigned c	har majorVersion									

unsigned char minorVe					
int bugfixVersion					
int buildNumber					
char array buildDate					

3.4	Retur til ind	dex						
		econda	ry clier	nt				
The followi	ng message	s have been	changed w	ith respect	to the prev	ious major :	software rel	ease 3.3
Please cons	sult SW/3 3 c	locumentati	ion for all o	ther nackag	AC			
r rease cons	Juit 3 W 3.3 C	ocamentati		ther packag	C3.			
Runtim	neExcer	otionMe	ssage					
int message								
unsigned cl	har message	eType = ROB	OT_MESSA	GE = 20				
uint64_t tir	***************************************							
char source		DODOT	NAECCA CE		EVERDEIGNI	40		
uint32 line		e = ROBOT_	_IVIESSAGE_	KUNTIME_I	EXCEPTION	= 10		
	ımnNumber							
	textMessage	:						

Retur til index							
Secondary cli	ent communication	ons inte	rface				
The secondary client co	mmunications interface is foun	d at TCP port	30002.				
Scheme							
4 bytes (int)	Length of overall package						
	- 1			 	 		
	Robot MessageType						
4 bytes (int)	Length of Sub-Package			 		<u></u>	<u> </u>
1 byte (uchar)	Package-Type						
n bytes	Content						
4 bytes	Length of Sub-Package						
1 byte	Package-Type			 	 		
n bytes	Content						
	: :			 			
I ength of overall package	 ge: total length of package incl	i			 		
Length of overall packa,	Ser total length of package men				 		
Robot MessageType: or	ly value 16 ("Robot State")						
Length of Sub-Package:	total length of subpackage incl	uding itself					
Package-Type:				 	 		
Possible types are:							

l	

	value 0 ("Robot Mo	de Data")						
	value 1 ("Joint Data	")						
	value 2 ("Tool Data'	')						
	value 3 ("Masterboa	ard Data")						
	value 4 ("Cartesian							
	value 5 ("Kinematic	s Info")						
	value 6 ("Configurat	ion Data")						
	value 7 ("Force Mod	de Data")						
	value 8 ("Additional	Info")						
	value 9 ("Calibratior	n Data")						
	value 10 ("Safety Da	ata")						
Robot State	package types							
							•••••	
Robot Mode	Data (value 0)							
Size in	Data type		Meaning		No			
bytes								
4	integer	Package length	(bytes)	46 byte				
1	uchar	Package type		Type is	0 ("Robot Mod	e Data")		
8	uint64	Timestamp						
1	boolean		Physical Robot Connected					
1	boolean		RealRobotEnabled					 ļ
1	boolean	RobotPowerOr						 ļ
1	boolean	EmergencyStop						 ļ
1	boolean	ProtectiveStop	ped					
1	boolean	ProgramRunnir	าต					

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_DISCO	NNECTED								
1	ROBOT_MODE_CONFI	RM_SAFETY								
2	ROBOT_MODE_BOOTI	ING								
3	ROBOT_MODE_POWE	R_OFF								
4	ROBOT_MODE_POWE	R_ON								
5	ROBOT_MODE_IDLE									
6	ROBOT_MODE_BACKE	DRIVE								
7	ROBOT_MODE_RUNN	ING								
8	ROBOT_MODE_UPDA	TING_FIRMWARE								
Control Mode	S									
Mode		Description	·	<u>'</u>		!				
0	CONTROL_MODE_POS	SITION								
1	CONTROL_MODE_TEA	кСН								
2	CONTROL_MODE_FOR	RCE								
3	CONTROL_MODE_TOF	RQUE								
Joint Data (va	alue 1)									
Size in	Data type	M	leaning							
bytes			,		0.7.1					
4	integer	Package length (byte	es)		251 bytes					
1	uchar	Package type				Joint Data"		. (6.)		
0	1. 11.						or each joir	it (6x):		
8	double	q actual			Actual join	•				
8	double	q target			Target join	-				
8	double	qd actual			Actual join					
	float	l actual			Actual joint current Actual joint voltage					
		V actual			-	_	ıro			
	float	T motor			Joint motor temperature Don't use, obsolete					
4	float	T micro			שסח't use,	opsolete				

1	uchar	Joint Mode	See table Joint Modes	
Joint Modes				
Mode (uchar)		Description		
236	JOINT_SHUTTING_DO	DWN_MODE		
237	JOINT_PART_D_CALI	BRATION_MODE		
238	JOINT_BACKDRIVE_N	MODE		
239	JOINT_POWER_OFF_	MODE		
245	JOINT_NOT_RESPON	DING_MODE		
246	JOINT_MOTOR_INITI	ALISATION_MODE		
247	JOINT_BOOTING_MO	DDE		
248	JOINT_PART_D_CALI	BRATION_ERROR_MODE		
249	JOINT_BOOTLOADER	_MODE		
250	JOINT_CALIBRATION	_MODE		
252	JOINT_FAULT_MODE			
253	JOINT_RUNNING_M	DDE		
255	JOINT_IDLE_MODE			
Tool Data (va	alue 2)			
Size in		Meaning	Notes	
bytes	Data type	ivicaning	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)	Descri	ption					
249	TOOL_BOOTLOADER_M	10DE					
253	TOOL_RUNNING_MOD	E					
255	TOOL_IDLE_MODE						
Masterboard							

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	ThreePositionEnal	olingDeviceInpu	ţ	From 3.2				
SafetyMode									
State (uchar)		Description					Comment		
1	SAFETY_MODE_NORM	1AL							
2	SAFETY_MODE_REDU	CED							
3	SAFETY_MODE_PROTECTIVE_STOP								
4	SAFETY_MODE_RECO	VERY							
5	SAFETY_MODE_SAFEG	GUARD_STOP		(SIO + SI1 +	SBUS) Phys	sical s-stop inte	rface input		
6	SAFETY_MODE_SYSTE	M_EMERGENCY_ST	ГОР	(EA + EB +	SBUS->Euro	map67) Physic	al e-stop inter	face input a	ctivated
7	SAFETY_MODE_ROBO	T_EMERGENCY_ST	OP	(EA + EB +	SBUS->Scre	en) Physical e-	stop interface i	input activat	ted
8	SAFETY_MODE_VIOLA	TION							
9	SAFETY_MODE_FAULT	-							
Cartesian Info	o (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	Υ	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?	CP offset, Rx-value (Rotation vector representation of TCP orientation	
8	double	TCPOffsetRY?	CP offset, Ry-value (Rotation vector representation of TCP orientation	
8	double	TCPOffsetRZ?	CP offset, Rz-value (Rotation vector representation of TCP orientation	

	· · · · · · · · · · · · · · · · · · ·
	,
on)	
on)	
on)	

Kinematics Info (value 5)

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

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'	

l	
	,

4	integer	MasterboardVersion?			
4	integer	ControllerBoxType?			
4	integer	RobotType?			
4	integer	RobotSubType?			
Force Mode D	ata (value 7)				
Size in	Data type	Meaning	Notes		
bytes					
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	Υ	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 Inf	o (value 8)				
Size in bytes	Data type	Meaning	Notes		
bytes 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 Da	ata (value 9)				
		ed internally by Universal Robots s	oftware only and should be skipped.		
It might be subject of	change in the near futu	ire.			
		<u> </u>			

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 ha	as a "source	e" code for	the sender	of the me	ssage. The	codes are:
Source is:						
68: Euromap 2						
67: Euromap 1						
66: Teach Pendant 2						
65: Teach Pendant 1						
30: Safety Processor I	3					
20: Safety Processor	4					
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 ar	e:					
16: ROBOT_STATE						
20: ROBOT_MESSAGE						
22: HMC_MESSAGE						
5: MODBUS_INFO_M	ESSAGE					
				•	-	-

Retur til index	
SAFETY_MODE_FAULT = 9;	
SAFETY_MODE_VIOLATION = 8	8;
SAFETY_MODE_ROBOT_EMER	RGENCY_STOP = 7;
SAFETY_MODE_SYSTEM_EME	RGENCY_STOP = 6;
SAFETY_MODE_SAFEGUARD_S	STOP = 5;
SAFETY_MODE_RECOVERY = 4	l;
SAFETY_MODE_PROTECTIVE_S	STOP = 3;
SAFETY_MODE_REDUCED = 2;	
SAFETY_MODE_NORMAL = 1;	

Retur til index			
MESSAGE_WARNING	_LEVEL_IN	FO = 1;	
MESSAGE_WARNING	_LEVEL_W	ARNING = 2	2;
MESSAGE_WARNING	_LEVEL_VI	OLATION =	3;
MESSAGE_WARNING	_LEVEL_FA	ULT = 4;	

Retur til index					
			l		
Up to	o softwai	re 3.2	From	softwar	e 3.3
NONE_VA	L = 0;		NONE_VA	L = 0,	
STRING_V	/AL = 3;		CONST_ST	RING_VAL	= 3,
LIST_VAL	= 4;		VAR_STRI	NG_VAL = 4	l,
POSE_VAI	L = 9;		LIST_VAL =		
BOOL_VA	L = 11;		POSE_VAL		
NUM_VA	L = 12;		BOOL_VAL		
INT_VAL =	= 13;		NUM_VAL		
FLOAT_V	AL = 14;		INT_VAL =	14,	
			FLOAT _VA	\L= 15,	

	: :	 		.	 		: -
Retur til index							
Example							
Lets say the client has recie	ved this data:						
Received 465 bytes.							
Bytes: [0] [0] [1] [209] [16] [[0] [0] [0] [29] [0] [0]	[0] [0] [0] [0] [5] <u>[</u>	[7] [246] [1] [1] [1	.] [0] [0] [0] [0] [63	3] [240] [0] [0] [0] [0]	[0] [0] [0] [0]	
[0] [251] [1] [64] [1] [78] [24		· · · · · · · · · · · · · · · · · · ·					
[0] [0] [66] [0] [102] [103] [6		· · · · · · · · · · · · · · · · · · ·	-			· 	[0] [C
[0] [0] [0] [0] [192] [1] [106]							
[210] [233] [51] [16] [35] [0]					***************************************		
[204] [104] [205] [239] [254	·····					· ··· ······	
[51] [66] [116] [204] [205] [2							
[157] [190] [66] [62] [0] [0]							
[191] [146] [242] [158] [148							
[217] [153] [52] [224] [36] [2		· · · · · · · · · · · · · · · · · · ·					
[176] [63] [243] [51] [88] [13						······	
[0] [0] [0] [0] [0] [64] [143] [
[14] [16] [225] [14] [17] [0]							
[241] [169] [253] [0] [0] [0]			-			· -)] [59]
[68] [155] [166] [66] [92] [0]							
So to start from the beginni	ing:					†	
[0],[0],[1],[209]: Length is 1	*256+209 = 465						
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	The state of the s						
[16] Package type is "Robot	State"						
3 7,72 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3							·
[0][0][0][29] First package le	ength is 29						
1-11-11-11-1	- 0						
[0] Package type is 0: "Robo	ot Mode Data"						
1 1 1 10 10 17 10 11 11 11 11 11							
	il Timestamp is 32971						
r 2 r-2 r-2 r-2 r-2 r-2 r-2 r-3 r-	,p =	:	:	<u> </u>	<u>: </u>	: :	

	l
1	
	Į l
	_

.					:		:		:
[1] is Dahat Cannastad True			 						
[1] isRobotConnected True							<u> </u>		<u> </u>
<u>:</u>			 				<u> </u>		<u></u>
							•		·
[1] isPowerOnRobot True			 						<u> </u>
[0] isEmergencyStopped False			 						
[0] isSecurityStopped False									
[0] isProgramRunning False							: :		
[0]			 		ļ		ļ		<u> </u>
[0] isProgrampaused False	<u> </u>		 <u>.</u>		<u>. </u>		<u> </u>		<u> </u>
[0] robot mode is 0									
<u>i</u> [63] [240] [0] [0] [0] [0] [0] [0]	Speed fraction as a	double	see http://	en.wikipedia.	org/wiki/Do	ouble-preci	sion_floati	i ng-point_fo	<u>i</u> ormat
[0] [0] [0] [251] Second packa	ge length is 251								
[0][0][0][202]	5				·				
[1] Second package type is "jo	int data"				<u> </u>				
[64] [1] [78][244] [77] [189] [2		sition							
[64] [1] [78] [247] [89] [95] [10		et nosition							
[01][1][70][217][03][33][1		et position			<u>.</u>		<u> </u>		<u></u>
[0] [0] [0] [0] [0] [0] [0] [0] [0]	nt 0 target speed						\$		
	rent		<u> </u>						
[[•	•	•	•	:	:	:
[66] [62] [0] [0] Joint 0 voltage			 						

	[
	
	,
	!
, network	byte order

[66] [0] [102] [10	3] Joint 0 r	notor temp	erature										
[66] [99] [153] [1	[66] [99] [153] [154] Joint 0 microprocessor temperature												
[253] Joint 0 joint	t mode												
[191] [246] [74] [170] [216]	[242] [29] [102] Joint 1	1 position									
etc													

Return to Index					
Meaning	Туре	Number o	Size in byt	Gnuplot co	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 vector
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 value
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 k

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 ε

		•		
r represent	ation of th	o tool orio	ntation	
represent	ation or th	ie toor one	iitatioii	
e of 5 corre	snonds to	hit 0 and h	it 2 set hig	h
2 01 3 00110	sporius to	Die o dila b	it 2 set mg	
oftware ve	rsion 1.8 a	nd on)		
bytes)				
entries at le	eading up t	he 756th b	yte.	

Return to Index						
Meaning	Туре	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 th	e 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 accelerat	ions
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 coor	dinates of the tool: (x,y,z,rx,ry,rz
TCP speed actual	double	6	48	62 - 67	Actual speed of the to	ool 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 coor	dinates of the tool: (x,y,z,rx,ry,rz
TCP speed target	double	6	48	80 - 85	Target speed of the to	ool given in Cartesian coordinate
Digital input bits	double	1	8	86	Current state of the d	igital 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 th	read execution time
Test value	double	1	8	94	A value used by Unive	ersal 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 Rol	oots software only
Tool Accelerometer values	double	3	24	109 - 111	Tool x,y and z acceler	ometer values (software version
	double	6	48	112 - 117	Used by Universal Rol	oots software only
Speed scaling	double	1	8	118	Speed scaling of the t	rajectory limiter
Linear momentum norm	double	1	8	119	Norm of Cartesian line	ear momentum
	double	1	8	120	Used by Universal Rol	oots software only
	double	1	8	121	Used by Universal Robots software only	

), where r	x, ry and rz	is a rotatio	n vector re	presentati	on of the t	ool orienta	tion	
s								
), where r	x, ry and rz	is a rotatio	n vector re	epresentati	on of the t	ool orienta	tion	
:S								
s encode	d as int64_t	., e.g. a valı	ue of 5 corr	esponds to	bit 0 and	bit 2 set hig	gh	
		_						
ontroller							*	
ontroller	(only fron	n software	version 1.8	and on)				
ontroller						1		
1.7)								

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

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	Туре	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 th	e 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 accelerat	ions
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 coor	dinates of the tool: (x,y,z,rx,ry,rz
TCP speed actual	double	6	48	62 - 67	Actual speed of the to	ool 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 coor	dinates of the tool: (x,y,z,rx,ry,rz
TCP speed target	double	6	48	80 - 85	Target speed of the to	ool given in Cartesian coordinate
Digital input bits	double	1	8	86	Current state of the d	ligital 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 th	read execution time
Test value	double	1	8	94	A value used by Unive	ersal 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 Ro	bots software only
Tool Accelerometer values	double	3	24	109 - 111	Tool x,y and z acceler	ometer values (software version
	double	6	48	112 - 117	Used by Universal Ro	bots software only
Speed scaling	double	1	8	118	Speed scaling of the t	rajectory limiter
Linear momentum norm	double	1	8	119	Norm of Cartesian lin	
	double	1	8	120	Used by Universal Ro	bots software only
	double	1	8	121	Used by Universal Ro	bots software only

), where rx, ry and	d rz is a rotati	on vector r	epresenta	tion of the	tool orient	ation	
S							
), where rx, ry and	d rz is a rotati	on vector r	epresenta ⁻	tion of the	tool orient	ation	
!S							
ts encoded as int6	4_t, e.g. a val	lue of 5 cor	responds t	o bit 0 and	bit 2 set h	igh	
						×	
ontroller			ا در داد در ۵			*	
ontroller (only f	rom software	e version 1.	8 and on)		1	7	
ontroller							
1 7\							
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

	4							
	•							
le enti	ries at leadi	ng up the	1060th byte	2.	•	•		
			=======================================	-		<u>i</u>		