ALEX30_GUI_DataOut GUI – Data OUT (from GUI to robot)

ALEX30_GUI_DataOut •
Host : ALEX30_GUI_DataOut_Host ÿÿ
Command : float

• Exos : ALEX30_GUI_DataOut_Exos
|_ Glo : ALEX30_GUI_DataOut_Exos_Global ÿÿ Command :
float |_ armRight :

ALEX30_GUI_DataOut_Exos_Arm |_ armLeft :

ALEX30_GUI_DataOut_Exos_Arm ÿÿ Command : float

const int ALEX32_COMMAND_EXOS_START_DEVICE = 1;

const int ALEX32_COMMAND_EXOS_STOP_DEVICE = 11;

const int ALEX32_COMMAND_EXOS_APPLY_JOINT_LIMIT = 50; ÿ applies the joint limits passed in Param_des.Joint_MinPos and Joint_MaxPos.

const int ALEX32_COMMAND_EXOS_APPLY_HUMAN_GRAVITY = 55; ÿ applies arm weight compensation using Param_des.Human_Arm_Gravity.

const int ALEX32_COMMAND_EXOS_STOP_HUMAN_GRAVITY = 56; ÿ disables arm weight compensation.

const int ALEX32_COMMAND_EXOS_APPLY_BILATERAL = 65; ÿ enable bilateral/ mirror mode using Param_des.Bilateral_factor.

const int ALEX32_COMMAND_EXOS_STOP_BILATERAL = 66; ÿ turns off bilateral/mirror mode.

const int ALEX32_COMMAND_EXOS_START_REHAB = 3; // ALEX32

const int ALEX32_COMMAND_EXOS_STOP_REHAB = 12;

const int ALEX32_COMMAND_EXOS_CLEARFAULT = 100;

ÿÿ Param_des : ALEX30_GUI_DataIn_Exos_Arm_Param

Joint_WearingPos[4] Joint positions (rad) for the arm “wearing” posture.

Joint_MinPos[4] Lower limits of the 4 actuated joints (in rad).
They are set by the GUI when you call applyRangeL/R (you work in degrees in the GUI and then convert).

Joint_MaxPos[4] Upper limits of the 4 joints (in rad).

X_Shulder_Offset Shoulder offset in meters along X

Human_Arm_Gravity Factor
[0,1] for the weight compensation of the human arm.
Used with commands:

or ALEX32_COMMAND_EXOS_APPLY_HUMAN_GRAVITY

or ALEX32_COMMAND_EXOS_STOP_HUMAN_GRAVITY

Bilateral_factor

Factor [0,1] for bilateral/mirror mode.
Used with:

or ALEX32_COMMAND_EXOS_APPLY_BILATERAL

or ALEX32_COMMAND_EXOS_STOP_BILATERAL

ALEX30_REHAB_DataOut – Data OUT (rehab commands/parameters ÿ robot)

ALEX30_REHAB_DataOut
ÿÿ Timer : float

ÿÿ armRight : ALEX30_REHAB_Exos_DataOut

ÿÿ armLeft : ALEX30_REHAB_Exos_DataOut
ÿÿ EE_Force_des[3] : float[] ÿÿ Desired force at EE (Fx, Fy, Fz), in what reference system?

Joint_Torque_des[4] : float[] ÿÿ Desired torque on the 4 actuated joints

EE_Pos_des[3] : float[] ÿÿ Desired position of the end-effector (x, y, z).

EE_Vel_des[3] : float[] ÿÿ Desired end-effector speed

EE_Impedance : Impedance_evo_str * ÿÿ EE position control – impedance parameters

EE_Speed_max : float ÿÿ EE position control: maximum handle speed

EE_Force_max : float ÿÿ EE position control: maximum handle force

Joint_Pos_des[4] : float[]

ÿÿ Joint_Vel_des[4] : float[]

ÿÿ Joint_Impedance1 : Impedance_str **

ÿÿ Joint_Impedance2 : Impedance_str

ÿÿ Joint_Impedance3 : Impedance_str

ÿÿ Joint_Impedance4 : Impedance_str

ÿÿ Joint_Speed_max[4] : float[]

ÿÿ Joint_Torque_max[4] : float[]

* Impedance_evo_str Impedance parameters to set when EE position control is active ?????

Impedance_evo_str
ÿÿPos : Impedance_base_str When is the error between desired and actual position positive?
ÿÿ Neg : Impedance_base_str When is the error between desired and actual position negative?
ÿÿ Revo[9] : float[] EE impedance rotation matrix

Impedance_base_str (3D, typically XYZ or 3-axis joint)

Impedance_base_str
ÿÿ K[3] : float[] Stiffness along the 3
ÿÿ C_rel[3] : float[] axes viscosity/damping coefficient along the 3 axes
ÿÿ C_ass[3] : float[] ¿ RELATIVE ? viscosity/damping coefficient along the 3 axes ¿ ABSOLUTE ?
ÿÿ Speed[3] : float[] Stiffness modification speed ??

** Impedance_str str ¿¿¿ Impedance parameters to set when Joint position control is active ?????

Impedance_str
: float
ÿÿK ÿÿ C_rel : float
ÿÿ C_ass : float
ÿÿ Speed : float

(struct that I read from the rehab memory segment (ALEX32_DATA_IN) and that in the code is mapped to AppDataInStruct.)

ALEX30_REHAB_DataIn

- Tmer : float

- armRight : ALEX30_REHAB_Exos_DataIn

- armLeft : ALEX30_REHAB_Exos_DataIn

Joint_Pos[8]	: float[]	8 joint positions
Joint_Speed[8]	: float[]	8 speed joint
Joint_Torque[4]	: float[]	4 couples
EE_Pos[3]	: float[]	3 EE positions (x,y,z)
EE_Speed[3]	: float[]	3 speed EE
EE_Force[3]	: float[]	3 EE forces
Joint_Pos_des_ret[4]	: float[]	4 desired joint positions
EE_Pos_des_ret[3]	: float[]	3 EE positions desired
Handle_Pressure	: float[]	1 knob pressure value

(struct that I read with the readGuiDataInStruct() command from the "ALEX32_GUI_IN" segment and which in the code is GuiDataInStruct.)

ALEX30_GUI_DataIn

- Host : ALEX30_GUI_DataIn_Host
  - Status : ALEX30_GUI_DataIn_Host_Status
    - Lib_FaultCode : Fault_Code
    - Connected : int
- Exos : ALEX30_GUI_DataIn_Exos
  - Glo : ALEX30_GUI_DataIn_Exos_Global
    - ÿÿ Status : ALEX30_GUI_DataIn_Exos_Global_Status
      - ÿÿ Fault Code : Fault_Code — Global exoskeleton fault
      - ÿÿ Rehab_Rec_DataOut : float
      - ÿÿ Control_Rec_DataOut : float
      - ÿÿ RecPlay_Rec_DataOut : float
      - ÿÿ CPU_Temperature : float — CPU Temperature
  - armRight : ALEX30_GUI_DataIn_Exos_Arm
  - armLeft : ALEX30_GUI_DataIn_Exos_Arm
    - Status : ALEX30_GUI_DataIn_Exos_Arm_Status
      - ÿÿ ControlPhase : float — current control phase (coded number).
      - ÿÿ ControlMode : float — control mode (impedance, etc..)
      - ÿÿ ToolMode : float — instrument mode (ALEX32).
      - ÿÿ Fault Code : Fault_Code
      - ÿÿ DriverBoard_FaultCode1: Fault_Code
      - ÿÿ DriverBoard_FaultCode2: Fault_Code
      - ÿÿ Driver_FaultCode1 : Fault_Code
      - ÿÿ Driver_FaultCode2 : Fault_Code
      - ÿÿ Driver_FaultCode3 : Fault_Code
      - ÿÿ Driver_FaultCode4 : Fault_Code
    - Param_curr : ALEX30_GUI_DataIn_Exos_Arm_Param
      - ÿÿ Joint_WearingPos[4] : float[] — wearing positions of the 4 actuated joints.
      - ÿÿ Joint_MinPos[4] : float[] — min joint current limit (in rad)
      - ÿÿ Joint_MaxPos[4] : float[] — min joint current limit (in rad)
      - ÿÿ X_Shulder_Offset : float — shoulder offset (m).
      - ÿÿ Human_Arm_Gravity : float ÿÿ — arm weight compensation factor [0 -1]
        Bilateral_factor : float — mirror control factor [0 -1]