STRUCT_DATA_OUT

ALEX30 GUI DataOut GUI-Data OUT (from GUI to robot)

ALEX30_GUI_DataOut

- Host : ALEX30_GUI_DataOut_Host
  - Command : float
- Exos : ALEX30_GUI_DataOut_Exos
  - Glo : ALEX30_GUI_DataOut_Exos_Global
    - Command : float
  - armRight : ALEX30_GUI_DataOut_Exos_Arm
  - armLeft : ALEX30_GUI_DataOut_Exos_Arm
    - 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;
→ applies the joint limits passed in Param des Joint MinPos and Joint MaxPos.

const int ALEX32_COMMAND_EXOS_APPLY_HUMAN_GRAVITY = 55;
+apply arm weight compensation using Param_des.Human_Arm_Gravity.

const int ALEX32_COMMAND_EXOS_STOP_HUMAN_GRAVITY = 56;
→disable arm weight compensation.

const int ALEX32 COMMAND_EXOS APPLY BILATERAL
-65; activate bilateral/mirror mode using Param des.Bilateral factor.

const int ALEX32_COMMAND EXOS_STOP BILATERAL
=66; 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;

- Param_des : ALEX30_GUI_DataIn_Exos_Arm_Param

Joint WearingPos[4] Joint positions (rad) for the arm "wearing" posture.

Jaint MinPos[4] Lower limits of the 4 actuated joints (in rad)
They are set by the GUI when you call applyRangeL/R (you in GUI work in degrees and pol 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 compensation of the weight of the human arm.
Used with commands:

- ALEX32_COMMAND_EXOS_APPLY_HUMAN_GRAVITY
- ALEX32_COMMAND_EXOS_STOP_HUMAN_GRAVITY

Bilateral_factor
Factor [0,1] for bitrate/mirror mode
Used with:

- ALEX32_COMMAND_EXOS_APPLY_BILATERAL
- ALEX32_COMMAND_EXOS_STOP_BILATERAL

STRUCT_DATA_OUT

ALEX30 REHAB DataOut-Data OUT (commands/parameters rehab + robot)

ALEX30_REHAB_DataOut
- Timer : float
- armRight : ALEX30_REHAB_Exos_DataOut
- armLeft : ALEX30_REHAB_Exos_DataOut
  - EE_Force_des[3] : float[] — Desired force at EE (Fx, Fy, Fa), in what reference system?
  - Joint_Torque_des[4] : float[] — Desired torque on the 4 actuated joints
  - EE_Pos_des[3] : float[] — Desired end-effector position (x, y, z)
  - EE_Vel_des[3] : float[] — Desired end-effector speed
  - EE_Impedance : Impedance_evo_str * — EE position control - impedance parameters
  - EE_Speed_max : float — EE position control: maximum handle speed
  - EE_Force_max : float — EE position control: maximum handle force
  - Joint_Pos_des[4] : float[]
  - Joint_Vel_des[4] : float[]
  - Joint_Impedance1 : Impedance_str **
  - Joint_Impedance2 : Impedance_str
  - Joint_Impedance3 : Impedance_str
  - Joint_Impedance4 : Impedance_str
  - Joint_Speed_max[4] : float[]
  - Joint_Torque_max[4] : float[]

"Impedance evo str ¿¿¿ Impedance parameters to set when EE position control is active ?????

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

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

Impedance_base_str
- K[3] : float[] — Stiffness along 13 asal
- C_rel[3] : float[] — viscosity/damping coefficient along the 3 axes — RELATIVE ?
- C_ass[3] : float[] — viscosity/damping coefficient along the 3 axes — ABSOLUTE?
- Speed[3] : float[] — Stiffness modification speed — ??

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

Impedance_str
- K : float
- C_rel : float
- C_ass : float
- Speed : float

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

STRUCT_DATA_IN

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 pairs
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 desired EE positions
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 GuiDatain Struct.)

STRUCT_DATA_IN

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