

ALEX30_GUI_DataOut GUI – Data OUT (dalla GUI verso il 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;

- └ → applica i limiti giunto passati in Param_des.Joint_MinPos e Joint_MaxPos.

- └ const int ALEX32_COMMAND_EXOS_APPLY_HUMAN_GRAVITY = 55;

- └ → applica la compensazione del peso braccio usando Param_des.Human_Arm_Gravity.

- └ const int ALEX32_COMMAND_EXOS_STOP_HUMAN_GRAVITY = 56;

- └ → disattiva la compensazione del peso braccio.

- └ const int ALEX32_COMMAND_EXOS_APPLY_BILATERAL = 65;

- └ → attiva la modalità bilaterale/mirror usando Param_des.Bilateral_factor.

- └ const int ALEX32_COMMAND_EXOS_STOP_BILATERAL = 66;

- └ → disattiva la modalità bilaterale/mirror.

- └ 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] Posizioni giunto (rad) per la postura di “indossaggio” del braccio.

- └ Joint_MinPos[4] Limiti inferiori dei 4 giunti attuati (in rad).

- └ Vengono impostati dalla GUI quando chiami applyRangeL/R (tu in GUI lavori in gradi e poi converti).

- └ Joint_MaxPos[4] Limiti superiori dei 4 giunti (in rad).

- └ X_Shoulder_Offset Offset spalla in metri lungo X

- └ Human_Arm_Gravity

- └ Fattore [0,1] per la compensazione del peso del braccio umano.

- └ Usato con i comandi:

- └ ○ ALEX32_COMMAND_EXOS_APPLY_HUMAN_GRAVITY

- └ ○ ALEX32_COMMAND_EXOS_STOP_HUMAN_GRAVITY

- └ Bilateral_factor

- └ Fattore [0,1] per la modalità bilaterale/mirror.

- └ Usato con:

- └ ○ ALEX32_COMMAND_EXOS_APPLY_BILATERAL

- └ ○ ALEX32_COMMAND_EXOS_STOP_BILATERAL

ALEX30_REHAB_DataOut – Data OUT (comandi/parametri rehab → robot)

ALEX30_REHAB_DataOut

- └ Timer : float
- └ armRight : ALEX30_REHAB_Exos_DataOut
- └ armLeft : ALEX30_REHAB_Exos_DataOut
 - └ EE_Force_des[3] : float[] Forza desiderata all'EE (Fx, Fy, Fz), in che sist. di riferimento ?
 - └ Joint_Torque_des[4] : float[] Coppia desiderata sui 4 giunti attuati
 - └ EE_Pos_des[3] : float[] Posizione desiderata dell'end-effector (x, y, z).
 - └ EE_Vel_des[3] : float[] Velocità desiderata dell'end-effector
 - └ EE_Impedance : Impedance_evo_str * EE position control – parametri di impedenza
 - └ 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 ??? Parametri di impedenza da impostare quando è attiva EE position control ?????

Impedance_evo_str

- └ Pos : Impedance_base_str ?? quando l'errore tra pos desiderata e attuale è positivo ??
- └ Neg : Impedance_base_str ?? quando l'errore tra pos desiderata e attuale è negativo ??
- └ Revo[9] : float[] EE impedance rotation matrix

Impedance_base_str (3D, tipicamente XYZ o 3 assi giunto)

Impedance_base_str

- └ K[3] : float[] Stiffness lungo i 3 assi
- └ C_rel[3] : float[] viscosità/coeff_smorzamento lungo i 3 assi ? RELATIVA ?
- └ C_ass[3] : float[] viscosità/coeff_smorzamento lungo i 3 assi ? ASSOLUTA?
- └ Speed[3] : float[] Stiffness modification speed ??

** Impedance_str str ??? Parametri di impedenza da impostare quando è attiva Joint position control ?????

Impedance_str

- └ K : float
- └ C_rel : float
- └ C_ass : float
- └ Speed : float

(struct che leggo dal segmento di memoria rehab (ALEX32_DATA_IN) e che nel codice è mappata in AppDataInStruct.)

ALEX30_REHAB_DataIn

- Tmer : float
- armRight : ALEX30_REHAB_Exos_DataIn
- armLeft : ALEX30_REHAB_Exos_DataIn
 - Joint_Pos[8] : float[] 8 posizioni giunto
 - Joint_Speed[8] : float[] 8 velocità giunto
 - Joint_Torque[4] : float[] 4 coppie
 - EE_Pos[3] : float[] 3 posizioni EE (x,y,z)
 - EE_Speed[3] : float[] 3 velocità EE
 - EE_Force[3] : float[] 3 forze EE
 - Joint_Pos_des_ret[4] : float[] 4 posizioni giunto desiderate
 - EE_Pos_des_ret[3] : float[] 3 posizioni EE desiderate
 - Handle_Pressure : float[] 1 valore di pressione manopola

(struct che leggo con comando readGuiDataInStruct() dal segmento "ALEX32_GUI_IN" e che nel codice è 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
 - └ Status : ALEX30_GUI_DataIn_Exos_Global_Status
 - └─ FaultCode : Fault_Code Fault globale dell'esoscheletro
 - └─ Rehab_Rec_DataOut : float
 - └─ Control_Rec_DataOut : float
 - └─ RecPlay_Rec_DataOut : float
 - └ CPU_Temperature : float Temperatura CPU
- armRight : ALEX30_GUI_DataIn_Exos_Arm
- armLeft : ALEX30_GUI_DataIn_Exos_Arm
 - Status : ALEX30_GUI_DataIn_Exos_Arm_Status
 - └─ ControlPhase : float fase di controllo corrente (numero codificato).
 - └─ ControlMode : float modalità di controllo (impedenza, etc..)
 - └─ ToolMode : float modalità strumento (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[] posizioni di wearing dei 4 giunti attuati.
 - └─ Joint_MinPos[4] : float[] limite min correnti dei giunti (in rad)
 - └─ Joint_MaxPos[4] : float[] limite min correnti dei giunti (in rad)
 - └─ X_Shulder_Offset : float offset spalla (m).
 - └─ Human_Arm_Gravity : float fattore di compensazione peso braccio [0 -1]
 - └─ Bilateral_factor : float fattore per il controllo mirror [0 -1]