UML

13 April 2023 13:40

Unidades : deg, mm, deg/s, mm/s, deg/s², mm/s²

Leyenda:

Tareas y secciones

(la letra es el identificador para que nos entendamos, vamos)

Interfaz en general (Para estandarizar las clases)

"clase"

* Constructor
* .begin(...) → para el setup (pinMode's esencialmente)
* resto de métodos

Con el mundo real (múltiples cpp y hpp)

- Pantalla
- Logger

⇝ (K)

Stepper_motor: (J) (I)

Crear base para que hereden las de abajo

- Arts → Posiciones
  - target
  - current
  - expected

Crear método watchdog para entrar en emergencia si aumenta la desviación (L)

↳ Rot (x2)
+ Stepper_motor
+ Sens_rotat.

(H)

Modelo M ⇓

q1 R
q2 P
q3 R
q4 R
1 efector

↳ Lin (x1) {carrera} → range
+ Stepper_motor
+ Sens_linear
+ Sens_homing

↳ Servo_feedback (x1)
+ Servo (<Servo.h>) Librería que ya existe para Arduino
+ Sens_rotat.
i_rel:=1

(G) (F)

- Actuador (hereda de Servo_feedback)

Contempla valores para saber si ha cogido hisopo, métodos open() y close() (E)

Código programa

Estados

- Uninit.
- Stop emergencia
- Homing required
- Swabbing_available
- Swabbing_procedure
- Swab_pick_up
- Swab_place

User input o ctes. código

User Input

(?)

q_clamp

"Para tomar y dejar hisopos dinámicamente, relegamos a funcs. dedicadas" ⇓ ~ procedimientos

Procedimiento/Movimiento

Contenedores de la STD a barajar: (D)
std::vector
std::forward_list

Clase movimiento: (C)
class pose
pos {x, y, z}
rot {rx, ry, rz}
*.hpp

* Kinematics.(hpp, cpp) Puede ser stand-alone o acompañar clase Robot (B)
↳ {q1_sens ... q_clamp, Pose target, current}

Cinemáticas

P.ej.:

Calculan

int Kin_dir(const q1_sens, const q2_sens, ..., &Pose (actual, p.ej.))
int Kin_inv(const &Pose, &q1 ...)

Calc.

Cods. error

* Cómputo en grados (deg_cmath.hpp, cpp) (A)

Funcs de cmath mezclazas con deg2rad y rad2deg

P.ej.: atan2 → atan2deg (double)

Hay que hacerlas

DPM si se usan templates, para float, double y long double

std::is_floating_point