

Read Me

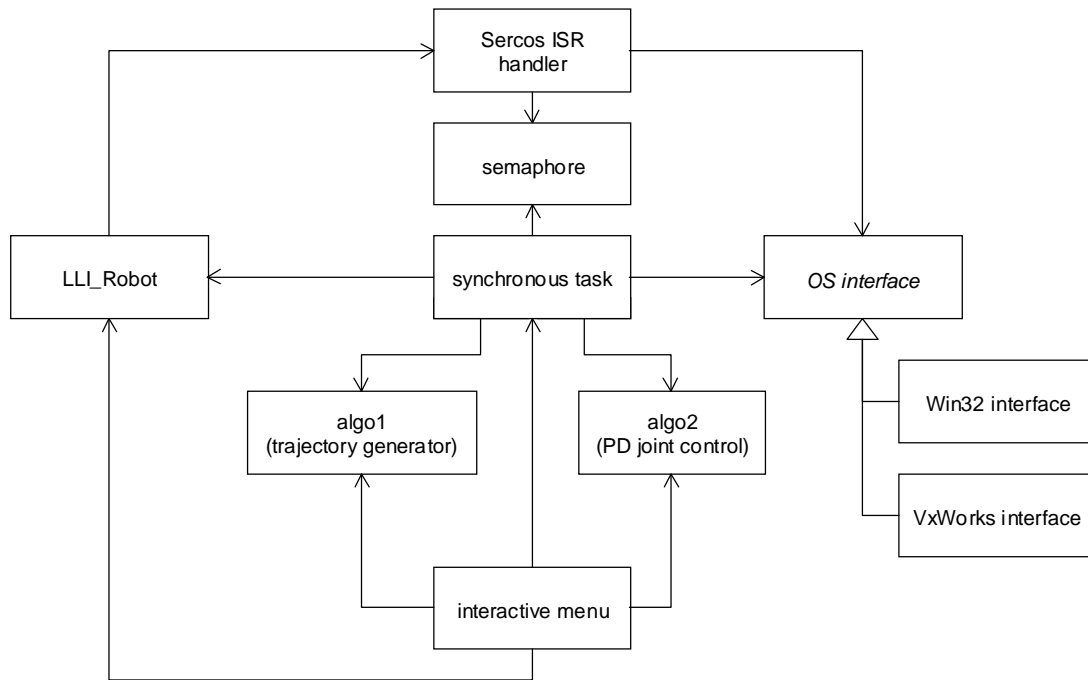
Product: LLI
LLI 6.6.4

Table of content

| | | |
|-------------|----------------------------------|----------|
| I. | OVERVIEW | 2 |
| II. | DETAILED DESCRIPTION..... | 3 |
| A. | LLI_ROBOT..... | 3 |
| B. | OS INTERFACE..... | 3 |
| C. | DRIVE INTERRUPT HANDLER..... | 3 |
| D. | ALGO 1 | 3 |
| E. | ALGO 2 | 3 |
| F. | SYNCHRONOUS TASK..... | 4 |
| G. | INTERACTIVE MENU..... | 4 |
| H. | TEST APPLICATION | 4 |
| III. | IMPLEMENTATION | 5 |

I. OVERVIEW

We use the UML formalism to describe the test structure, even if implementation is done in C. In the following, the term « class » does not refer to a C++ class (implementation), but to a set of functions and associated data (design).



II. DETAILED DESCRIPTION

a. LLI_Robot

LLI_robot is provided by the LLI C interface. It calls regularly a plug-in function: the drive interrupt handler. It provides a set of functions. Synchronous functions are used by the synchronous task, whereas asynchronous functions are used by the test application (which is an asynchronous task).

b. OS interface

The test can be compiled for VxWorks or Window. The OS interface provides a common interface to these two operating systems. It is an abstract class, which means it is only an interface definition, and it contains no code. The classes which need to access operating system functions access them through this general interface. The methods provided are:

- create a binary semaphore
- give a binary semaphore
- take a binary semaphore
- spawn a task
- delete a task

The implementations of this interface for Windows and VxWorks are provided by the derived classes « Win32 interface » and « VxWorks interface », respectively. These two classes share the same interface defined by « OS interface ».

c. Drive interrupt handler

LLI_Robot call this handler within the context of the drive interrupt (each 4 ms for Sercos drives, possibly faster for Starc drives). The handler gives a semaphore at each interrupt. Two methods are provided:

- configuration : define which semaphore the IRS handler should give
- a function pointer (LLI_Plugin) to be linked to LLI_Robot

d. Algo 1

It consists in a very simple movement generator used to compute the commands. It should be used when the robot is in standard position control mode.

The following methods are provided:

- initialize algorithm
- compute the new setpoint at each cycle time
- start a movement on a joint

e. Algo 2

It consists in a very simple position controller (proportional). It should be used when the robot is in torque control mode.

The following methods are provided:

- initialize algorithm
- compute the new setpoint at each cycle time
- set/get gains

WARNING! Invalid gains may lead to robot instability, unsafe robot movements, damage to the robot, and injury to the user.

f. Synchronous task

This task is pending on the synchronization semaphore given by the Sercos ISR handler. At each cycle time, it reads the LLI_Robot feedbacks, and sets the new command. The command is compute by algo1 or algo2.

The following methods are provided

- start the synchronous task
- stop the synchronous task
- get current commands
- get current feedback
- set/get the current algorithm (1 or 2)

g. Interactive menu

It is a text user interface (menus) which enables to access:

- each asynchronous LLI_Robot function,
- synchronous function parameters (commands and feedbacks)
- parameters for algo1 (joint to be moved)
- parameters for algo2 (gains)

h. Test application

This is the main task of the test. It manages the following aspects:

- construct application, link objects together, and initialize the system.
- run user interface until exit
- manage application end, stop tasks, and destroy all objects

III. IMPLEMENTATION

This test is implemented in C. LLI_Robot is provided by the LLI library.

| File | Description |
|-------------------|--|
| Systemfunctions.h | Defines the interface for using « OS interface » and « Semaphore », independently of the OS (Windows or VxWorks) |
| WinFunctions.c | Implementation of the functions defined in Systemfunctions.h specific for Windows. This file is compiled only under Windows. |
| VxFunctions.c | Implementation of the functions defined in Systemfunctions.h specific for VxWorks. This file is compiled only under VxWorks. |
| Isr.h | Interface of Sercos ISR handler |
| Isr.c | Implementation of Sercos ISR handler |
| Synchro.h | Interface of the synchronous task |
| Synchro.c | Implementation of the synchronous task |
| Algo1.h | Interface of algo1 |
| Algo1.c | Implementation of algo1 |
| Algo2.h | Interface of algo2 |
| Algo2.c | Implementation of algo2 |
| _test.h, text.h | Header for test applications |
| Test.c | Test application and interactive menu for robot control |
| TestKin.c | Test application and interactive menu for robot kinematics |